تعلم الآلة وعلم البيانات

الأساسيات والمفاهيم والخوارزميات والأدوات

تأليف: ميلاد وزان

ترجمة: د. علاء طعيمة



بسمه تعالى

تعلم الآلة وعلم البيانات

الأساسيات والمفاهيم والخوارزميات والأدوات

تأليف:

ميلاد وزان

ترجمة:

د. علاء طعيمة

مقدمة المترجم

يقدم هذا الكتاب دروساً تعليمية عديدةفي مجال تعلم الالة وعلم البيانات ويتميز الكتاب ببساطة لغته وسهولة فهمها من قبل القارئ مع شرح مميز مدعم بالأمثلة والتمارينفي نهاية كل فصل.

عند انتهائي من قراءه هذا الكتاب، احببت ان اترجم هذا الكتاب وبعد التواصل مع المؤلف الاستاذ ميلاد وزان لم يبد مانعا من ترجمته الى العربية. لقد اخترت كتاب "يادگيرى ماشين وعلم داده ها" للأستاذ ميلاد وزان لما رأيته من جودة هذا الكتاب، وللمنهجية التي اتبعها المؤلففي ترتيبه وبساطة شرحه. لقد حاولت قدر المستطاع ان اخرج بترجمة ذات جودة عالية، ومع هذا يبقى عملاً بشرياً يحتمل النقص، فاذا كان لديك أي ملاحظات حول هذا الكتاب، فلا تتردد بمراسلتنا عبر بريدنا الالكتروني alaa.taima@qu.edu.iq.

نأمل ان يساعد هذا الكتاب كل من يريد ان يدخلفي مجال تعلم الالة وعلم البيانات ومساعدة القارئ العربي على تعلم هذا المجال. اسأل الله التوفيقفي هذا العمل لأثراء المحتوى العربي الذي يفتقر أشد الافتقار إلى محتوى جيد ورصينفي مجال الذكاء الاصطناعي وتعلم الالة والتعلم العميق وعلم البيانات. ونرجو لك الاستمتاع مع هذا الكتاب ولا تنسونا من صالح الدعاء.

د. علاء طعيمة

كلية علوم الحاسوب وتكنولوجيا المعلومات

جامعة القادسية

العراق

مقدمة المؤلف

علم البيانات وتعلم الآلة

اليوم، إذا جمعت أبرز قادة الأعمال العالميين وطلبت منهم إحداث أكبر فرق بين الأعمال التجارية في القرنين العشرين والحادي والعشرين ، فمن المرجح أن يقولوا نفس الشيء: **البيانات**.

منذ مطلع القرن، نمت كمية البيانات بمعدل مذهل مع ظهور وسائل التواصل الاجتماعي ، والهواتف الذكية ، وإنترنت الأشياء ، والتطورات التكنولوجية الأخرى. تشير التقديرات إلى أن أكثر من 90٪ من جميع البيانات التي تم إنشاؤها بواسطة الإنسان قد تم إنشاؤها في السنوات الخمس الماضية. يُعرف هذا الانفجار المعلوماتي باسم "**البيانات الضخمة**" وسيغير العالم من حولنا تمامًا. أدى النمو الهائل في إنتاج البيانات إلى جعل المؤسسات أكثر اهتمامًا من أي وقت مضى بكيفية استخدام البيانات لتعزيز اهتماماتها التجارية. وفي الوقت نفسه ، يتطلع الأشخاص بشكل متزايد إلى تطوير مهاراتهم في البيانات لتسليط الضوء على السيرة الذاتية والتقدم الوظيفي والأمن الوظيفي.

تعد البيانات اليوم أداة ووقودًا للشركات لاكتساب رؤى مهمة وتحسين أدائها. سيطر علم البيانات على كل صناعة تقريبًا في العالم. لا توجد صناعة في العالم اليوم لا تستخدم البيانات. لكن من سيحصل على هذه البصيرة؟ من يقوم بمعالجة جميع البيانات الخام؟ كل شيء يقوم به محلل بيانات أو عالم بيانات. هذان هما أكثر الأدوار الوظيفية شيوعًا في هذا المجال. لأن الشركات حول العالم تتطلع إلى تحقيق أقصى استفادة من بياناتها. بالنسبة للأشخاص الذين يبحثون عن فرص عمل طويلة الأجل ، لطالما كانت وظائف علوم البيانات خيارًا عمليًا وموثوقًا به. من المرجح أن يستمر هذا الاتجاه مع دمج الذكاء الاصطناعي والتعلم الآلي في حياتنا اليومية واقتصادياتنا.

البيانات لا تخبرنا فقط عن الماضي. إذا قمنا بنمذجة البيانات بعناية وبدقة ، يمكننا العثور على أنماط وارتباطات للتنبؤ بسوق الأوراق المالية ، وانتاج تسلسل البروتين ، واكتشاف الهياكل البيولوجية مثل الفيروسات ، وأكثر من ذلك. ومع ذلك ، فإن نمذجة كميات كبيرة من البيانات يدويًا أمر شاق. لتحقيق ذلك ، لجأنا إلى خوارزميات التعلم الآلي التي يمكن أن تساعدنا في استخراج المعلومات من البيانات. يؤدي التعلم الآلي للأتمتة المهام التي تتطلب عادةً ذكاءً بشريًا. يمكن تعريف التعلم الآلي ، بدوره ، على أنه استخدام وتطوير أنظمة الكمبيوتر القادرة على التعلم والتكيف دون تخطيط صريح. تستخدم هذه الأنظمة الخوارزميات والنماذج الإحصائية لتحليل واستنتاج الأنماط في البيانات. يمكن لخوارزميات التعلم الآلي أن تحل المشكلات المعقدة غير العملية أو التي يستحيل القيام بها يدويًا ، وتعلم التوزيعات والأنماط والارتباطات لكشف المعرفة داخل البيانات. تقوم الخوارزميات بذلك عن طريق استكشاف مجموعة بيانات وإنشاء نموذج

تقريبي لتوزيع البيانات ، بحيث عندما نقوم بتغذية بيانات جديدة وغير مرئية ، فإنها ستؤدي إلى نتائج جيدة.

بشكل عام ، يمكن أن تحقق خوارزميات التعلم الآلي القدرة على التعلم من خلال ثلاث طرق مختلفة:

- **التعلم مع الاشراف:** التعلم الخاضع الاشراف في التعلم الآلي هو طريقة تنشئ نموذجًا للتنبؤ بالنتائج بناءً على البيانات المصنفة. يعني وجود البيانات المصنفة أنه يتم تقديم إجابة أو حل لكل عينة من مجموعة البيانات. كمثال بسيط ، إذا أردنا أن يقوم نموذج التعلم الآلي الخاص بنا بعمل تنبؤات حول فاكهة التفاح أو الموز ، فقم بتسمية القيم "تفاحة" أو "موز" بمجموعة من السمات مثل الوزن والطول والعرض وأي قياسات ذات صلة من الفاكهة المتاحة. دعونا نلقي نظرة على مثال أكثر صلة بالأعمال التجارية؛ خسارة العملاء. لفهم رفض العميل بشكل أفضل ، يجب عليك أولاً تحليل المقاييس التي قد تؤدي إلى خروج العميل. تتضمن مجموعة البيانات الخاصة بك لهذا النوع من النماذج متغيرات الفهرس مثل الأيام المنقضية منذ آخر عملية شراء ، ومتوسط مبلغ الشراء ، ومتغير التوقع المسمى ، سواء كان الشخص لا يزال عميلاً أم لا. نظرًا لأن لدينا بيانات سابقة عن حالة العميل ، فإن إنشاء نموذج باستخدام هذا النوع من مجموعة البيانات يمكن أن يكون مرشحًا رائعًا للتعلم الخاضع للإشراف.
- **التعلم بدون اشراف:** التعلم غير الإشرافي في التعلم الآلي هو عندما نقدم أمثلة إلى الخوارزمية دون أي توجيه ونترك إنشاء التسمية للخوارزمية. بمعنى آخر ، يسعى التعلم غير الإشرافي إلى العثور على أنماط مخفية في البيانات غير المسماة وإنشاء مجموعات وكلسترات. على سبيل المثال ، عندما نحتاج إلى فهم كيفية تصنيف المجموعات ضمن مجموعة بيانات العميل إلى أقسام متشابهة بناءً على خصائصها وسلوكياتها. غالبًا ما يستخدم التعلم غير الإشرافي للتحليل الاستكشافي وتشخيص التشوهات. لأنه يساعد في معرفة كيفية ارتباط أجزاء البيانات والاتجاهات التي قد توجد. يمكن استخدامها للمعالجة المسبقة لبياناتك قبل استخدام خوارزمية التعلم الخاضع للإشراف.
- **التعلم المعزز:** التعلم المعزز هو أسلوب يوفر تغذية راجعة تعليمية باستخدام آلية المكافأة. تحدث عملية التعلم كعامل يتفاعل مع البيئة ويحاول طرقًا مختلفة لتحقيق نتيجة. يتلقى الوكيل مكافأة أو عقوبة عندما يصل إلى الهدف أو لا. من خلال ملاحظات التعلم هذه ، يتعلم الوكيل أي المواقف تؤدي إلى نتائج جيدة والتي تؤدي إلى الفشل ويجب تجنبها. على سبيل المثال ، يمكننا استخدام التعلم المعزز عندما نحتاج إلى برنامج للعمل بنجاح في بيئة تنافسية ، مثل لعبة الفيديو أو سوق الأوراق المالية. في هذه الحالة ، يبدأ البرنامج بالعمل في البيئة ويتعلم مباشرة من أخطائه حتى

يجد مجموعة من القواعد التي تضمن نجاحه. لا يتطلب التعلم المعزز بيانات مصنفة بالإضافة إلى التعلم الخاضع للإشراف. بالإضافة إلى ذلك ، فإنه لا يستخدم حتى مجموعة بيانات غير مسماة مثل التعلم غير الخاضع للإشراف. التعلم المعزز ، بدلاً من محاولة اكتشاف علاقة في مجموعة بيانات ، يتم تحسينه باستمرار بين نتائج تجارب الفرد السابقة وكذلك إنشاء تجارب جديدة. بمعنى آخر ، فإنه ينشئ مجموعات بيانات جديدة ونتائج مع كل جهد.

التعلم الخاضع للاشراف مهم جداً جداً في علم البيانات ، لأنه يسمح لنا بالقيام بما يتوق إليه الجنس البشري: **التنبؤ**. التنبؤ مفيدة جداً في الأعمال التجارية وللربح ، وهي تمكننا من القيام بأفضل ما في وسعنا ، لأننا من خلال التنبؤ نعرف النتيجة المحتملة لأي موقف.

قد يبدو التعلم الخاضع للاشراف بمثابة سحر للبعض. ومع ذلك ، فإن التعلم الخاضع للإشراف ليس سحراً بأي حال من الأحوال. بدلاً من ذلك ، يساعد التعلم الخاضع للاشراف على أساس الإنجازات البشرية في الرياضيات والإحصاء وباستخدام الخبرات والملاحظات البشرية وتحويلها إلى تنبؤات دقيقة بطريقة لا يستطيع أي عقل بشري القيام بها.

ومع ذلك ، لا يمكن التنبؤ بالتعلم الخاضع للإشراف إلا في ظل ظروف مواتية معينة. لذلك ، من أجل القيام بذلك ، من المهم جدًا الحصول على أمثلة من الماضي يمكننا من خلالها استنباط القواعد والنقاط التي يمكن من خلالها إجراء تنبؤ محتمل للغاية.

لماذا تعلم الالة؟

كمية البيانات المتاحة لنا تتزايد باستمرار. تستخدم الآلات هذه البيانات للتعلم وتحسين النتائج وتزويدنا بها. يمكن أن تكون هذه النتائج مفيدة للغاية في تقديم رؤى قيمة وكذلك اتخاذ قرارات تجارية مستنيرة. يتطور التعلم الآلي باستمرار ، ونتيجة لذلك ، تنمو تطبيقات التعلم الآلي أيضًا. نحن نستخدم التعلم الآلي أكثر مما نعرفه في حياتنا اليومية. لقد دخل التعلم الآلي في حياتنا اليومية ، حتى بدون أن ندرك ذلك. لقد مكنت خوارزميات التعلم الآلي العالم من حولنا. ويمكن القول أن المستقبل هنا بالفعل ، ويلعب التعلم الآلي دورًا مهمًا في تشكيل أفكارنا المعاصرة.

اليوم ، يحظى التعلم الآلي بكل الاهتمام الذي يحتاجه. يمكن أن يؤدي التعلم الآلي العديد من المهام تلقائيًا ، خاصة تلك التي لا يمكن إلا للبشر القيام بها بذكائهم الفطري. لا يمكن إعادة إنتاج هذا الذكاء في الآلات إلا بمساعدة التعلم الآلي.

بمساعدة التعلم الآلي ، يمكن للشركات أتمتة المهام الروتينية. كما أنه يساعد على أتمتة وإنشاء نماذج لتحليل البيانات. تعتمد الصناعات المختلفة على كميات كبيرة من البيانات لتحسين أدائها واتخاذ قرارات ذكية. يساعد التعلم الآلي على إنشاء نماذج يمكنها معالجة وتحليل كميات كبيرة من البيانات المعقدة وتقديم نتائج دقيقة. هذه النماذج دقيقة وقابلة للتطوير وتعمل بوظيفة ذات

وقت أقل. من خلال بناء مثل هذه النماذج الدقيقة للتعلم الآلي ، يمكن للشركات الاستفادة من الفرص المربحة وتجنب المخاطر غير المعروفة.

يتم استخدام التعرف على الصور وتوليد النص وتصنيف النص وتشخيص الأمراض وغيرها الكثير في العالم الحقيقي. ومن ثم ، فإنه يزيد المجال لخبراء التعلم الآلي للتألق كمحترفين. بالإضافة إلى ذلك ، تتخلف العديد من الشركات في مجال التكنولوجيا بسبب الوتيرة السريعة للقفزات التقنية. يعد التحول الرقمي صناعة ضخمة ، وحقيقة الأمر هي أنه لا يوجد عدد كافٍ من خبراء التعلم الآلي لتلبية احتياجات الصناعة الجديدة.

إذا كنت ترغب في الارتقاء بحياتك المهنية إلى المستوى التالي ، فيمكن للتعلم الآلي أن يفعل ذلك نيابةً عنك. إذا كنت تتطلع إلى الانخراط في شيء يجعلك جزءًا من شيء عالمي ومعاصر ، فيمكن للتعلم الآلي أن يفعل ذلك من أجلك.

العلاقة بين تعلم الآلة وعلم البيانات

يمكن أن يوفر التعلم الآلي رؤى قيمة فقط إذا تلقى بيانات جيدة. وبالتالي ، بدون استخدام بيانات نظيفة ومتسقة وعالية الجودة ، يمكن إنشاء رؤى ذات مغزى كمي (إن وجدت). في الوقت نفسه، يحتاج عالم البيانات إلى التعلم الآلي ، لأنه يكاد يكون من المستحيل فهم نتائج الكمية الهائلة من البيانات المعقدة التي تمتلكها المؤسسات والتنبؤ بها بدقة.

يجب أن يتمتع عالم البيانات أيضًا بحس تجاري كبير لفهم ما يجعل الأعمال التجارية ناجحة، وأين يمكن تحسينها ، وما هي الخيارات الممكنة لتحقيق هذه النتيجة. يمكن اعتبار التعلم الآلي بمثابة إحصاء عملي للقيام بذلك. بمعنى آخر ، التعلم الآلي هو تكامل علوم الكمبيوتر والتخصصات الرياضية المختلفة التي تُستخدم فيها مفاهيم علوم الكمبيوتر لبناء نماذج رياضية قوية يمكنها حل مجموعة من المشكلات المتشابهة وذات الصلة.

يهتم علماء البيانات بضمان أن نموذج التعلم الآلي يحقق أهداف المشروع. هذا هو المكان الذي تكون فيه مجموعة مهارات العمل في متناول اليد. للنجاح في تطوير نموذج التعلم الآلي ، يجب أن يكون لدى عالم البيانات فهم معقول للمشكلة المطروحة وأهداف المشروع. بدون هذا، هناك فرصة ضئيلة للنجاح لأي برنامج علم البيانات ونموذج التعلم الآلي. في هذا الصدد ، يقضي ما يقرب من 80٪ من وقت عالم البيانات في استكشاف البيانات وتنظيفها وإعدادها. القيام بذلك بشكل صحيح هو جزء أساسي من العملية. بمجرد الانتهاء ، يمكن لعالم البيانات البدء في تطوير نموذج التعلم الآلي. يمكنهم اختبار ومقارنة النماذج المختلفة ثم تحسين المرشح الواعد لبيئة الإنتاج.

واحدة من أكثر الطرق فعالية لمراقبة قبول هذه النماذج هي من خلال التمثيل المرئي للبيانات. يتيح تقديم تقرير بالبيانات لقادة الأعمال اتخاذ قرارات مستنيرة يمكن أن تفيد المنظمة. بصرف

النظر عن توفير البيانات ، ربما يكون هذا التمثيل المرئي للبيانات هو أهم خطوة في المساعدة على ضمان نجاح المشروع.

وبالتالي ، في حين أن نماذج التعلم الآلي مهمة ، فإن نجاحها يعتمد بشكل كبير على قدرة فريق البيانات على فهم البيانات المنظمة وتوفيرها بمعلومات ممتازة تسمح للنموذج بعمل تنبؤات دقيقة.

علم البيانات والتعلم الآلي مترابطان وأساسيان لنجاح أي مؤسسة قائمة على البيانات. ومع ذلك ، كل هذا يتوقف على جودة البيانات المستخدمة.

عن الكتاب

يتكون الكتاب الحالي من جزئين: **المقدمة والتعلم الآلي** ، ويتكون من 9 فصول. توجد معظم محتويات الكتاب الرئيسية والمهمة في الجزء الثاني. لذلك ، إذا كنت معتادًا على مفاهيم البرمجة ومعالجة البيانات ، فإن الجزء الأول من الكتاب غير مناسب لك ويمكنك تخطي هذا الجزء والدخول إلى قسم التعلم الآلي مباشرة.

القراء المستهدفين

يمكن اعتبار هذا الكتاب بمثابة مقرر دراسي اختياري لطلاب السنة النهائية الجامعيين وكتاب منهجي لطلاب الدراسات العليا في الهندسة وعلوم الكمبيوتر في تخصص الذكاء الاصطناعي. يمكن أن يكون هذا الكتاب أيضًا مرجعًا جيداً لجميع المهتمين بالتعلم الآلي وعلم البيانات ، من الباحثين في مختلف المجالات إلى الأطباء.

القراء الأعزاء، يمكنكم إخباري بأي انتقادات أو اقتراحات أو إذا رأيتم أي مشاكل في الكتاب:

vazanmilad@gmail.com

ميلاد وزان

الحادي والعشرون من كانون الثاني (يناير) ألف وأربعمائة ــ خراسان رضوى، كاشمر

" ينبغي أن نبسّط كل شيء بقدر الامكان ، ولكن دون أن نفرط في تبسيطه "

آلبرت اينشتاين

المحتويات

الجزء الثاني

الجزء الأول

المقدمه

يشمل الفصول التالية:

الفصل الأول: علم البيانات

الفصل الثاني: مقدمة في بايثون

الفصل الثالث: البيانات

1 علم البيانات

الأهداف:

- ما هو علم البيانات؟
- العلاقة بين علم البيانات والذكاء الاصطناعى والبيانات الضخمة.
- الفرق بين محلل البيانات ومهندس وعالم البيانات.
- التعرف على المهارات اللازمة لتصبح عالم بيانات.
- أهمية علم البيانات وتطبيقه.

ما هو علم البيانات؟

في القرن الماضي ، كان النفط يعتبر بمثابة الذهب الأسود ، ومع الثورة الصناعية وظهور صناعة السيارات ، أصبح النفط القوة الدافعة الرئيسية للحضارة الإنسانية. ومع ذلك ، مع مرور الوقت ، انخفضت قيمتها بسبب توافر مصادر الطاقة المتجددة البديلة. في القرن الحادي والعشرين ، القوة الدافعة الجديدة هي البيانات والتي تؤثر على الصناعات. تُظهر هذه البيانات وتحيط بكل جانب من جوانب كل ما نعرفه ونفعله. حتى صناعة السيارات تستخدم البيانات لضمان استقلالية وسلامة سياراتهم.

اليوم ، يمكن اعتبار علم البيانات بمثابة القوة الكهربائية التي تمنح القوة للصناعات وقد أحدثت ثورة كبيرة في جميع الصناعات تقريبًا. تحتاج الصناعات إلى البيانات لتحسين أدائها وزيادة نمو أعمالها وتقديم منتجات أفضل لعملائها. ومع ذلك ، بينما تعتمد معظم المؤسسات على البيانات لتوجيه قرارات الأعمال ، فإن البيانات وحدها ليست الهدف. إذا تعذر الحصول على الأفكار القيمة التي تؤدي إلى إجراءات أكثر استنارة ، فإن الحقائق والأرقام لا معنى لها.

يتطور علم البيانات كمسار وظيفي واعد. كما قد تكون سمعت ، أدخلت جامعة هارفارد المهنة الرائعة للقرن الحادي والعشرين كعالم بيانات. علم البيانات هو مستقبل الذكاء الاصطناعي ومستقبل الذكاء المهني. أصبحت الصناعات تعتمد على البيانات وتحتاج إلى علماء بيانات لمساعدتها على اتخاذ قرارات أكثر ذكاءً وإنشاء منتجات مهمة. في عالم اليوم أيضًا ، أصبحت الحاجة إلى معرفة البيانات أمرًا ضروريًا. نحتاج أن نتعلم كيف يمكن تحويل البيانات الأولية إلى منتجات ذات مغزى. نحن بحاجة إلى تعلم التقنيات وفهم متطلبات تحليل الرؤى من البيانات. البيانات لديها إمكانات غير مستغلة يجب تحقيقها من أجل تطوير منتجات مفيدة. مع ظهور تقنيات التعلم الآلي والتعلم العميق ، أصبح التنبؤ بالبيانات الذكية وتصنيفها ممكنًا.

البيانات الضخمة وعلم البيانات هي مفتاح المستقبل. لذلك من المهم جداً معرفة ما هو علم البيانات وكيف يمكن أن يؤثر على أعمالنا والبيئة من حولنا. نحن بحاجة إلى أن نكون ديناميكيين، وأن نعمل مع التكنولوجيا ونتقدم إلى الأمام ونتعلم علم البيانات قبل فوات الأوان.

تعريف علم البيانات

علم البيانات هو مجال بحث يكتشف من خلال مجموعةً من الأدوات والخوارزميات ومبادئ التعلم الآلي المختلفةً والتنقيب عن النصوص والرياضيات والإحصاءات عن الأنماط المخفيةً للبيانات الخام.

العاملين في مجال علم البيانات ، من خلال تطبيق خوارزميات التعلم الآلي والرياضيات والإحصاءات على أنواع مختلفة من البيانات ، يحاولون بناء نظام ذكاء اصطناعي لأداء المهام التي تتطلب عادةً ذكاءً بشريًا ، أو لتقسيم المشكلات المعقدة إلى مشاكل أصغر لاكتساب الأفكار والمعرفة منها. في المقابل ، تولد هذه الأنظمة رؤية تلعب دورًا مهمًا للغاية في تعزيز أهداف

المحللين والمستخدمين التجاريين. باختصار ، **الغرض من علم البيانات** هو شرح العمليات من خلال البيانات الموجودة. من المتوقع أن يكون هذا التفسير دقيقًا بما يكفي لعمل تنبؤات. الهدف النهائي من هذه التفسيرات والتوضيحات هو اتخاذ قرارات مستنيرة بناءً على المعرفة المستخرجة من هذه البيانات.

ما هي البيانات الضخمة؟

البيانات هي أساس علم البيانات ؛ البيانات هي المكونات الرئيسية التي تستند إليها جميع التحليلات. في مجال علم البيانات ، يمكن تقسيم هذه البيانات إلى مجموعتين: **البيانات التقليدية والبيانات الضخمة**.

تشير البيانات التقليدية إلى البيانات المخزنة في قواعد البيانات التي يمكن للمحللين إدارتها على جهاز الكمبيوتر. هذه البيانات في شكل جدول يحتوي على قيم عددية أو نصية. بالطبع ، مصطلح "تقليدي" هو ما نستخدمه في أغلب الأحيان للتمييز بشكل أفضل بين البيانات الضخمة وأنواع البيانات الأخرى. البيانات الضخمة ، من ناحية أخرى ، هي بيانات أكبر من البيانات التقليدية وعادة ما يتم توزيعها عبر شبكة واسعة من أجهزة الكمبيوتر.

تعريف البيانات الضخمة

تشير البيانات الضخمة إلى مجموعة بيانات هيكلية معقدة وغير منظمة وذات حجم كبيريتم إنشاؤها بسرعة من مجموعة متنوعة من المصادر، مما يزيد من التبصرواتخاذ القرار.

تشير البيانات الضخمة إلى مجموعة كبيرة من البيانات غير المتجانسة التي يتم الحصول عليها من مجموعة متنوعة من المصادر وتتضمن أنواعًا مختلفة من البيانات على النحو التالي:

- **البيانات غير المهيكلة:** الشبكات الاجتماعية ، ورسائل البريد الإلكتروني ، والمدونات ، والتغريدات ، والصور الرقمية ، وبيانات الجوال ، وصفحات الويب ، إلخ.
- **شبه المنظمة:** ملفات XML ، ملفات نصية ، إلخ.
- **البيانات المهيكلة:** قواعد البيانات والتنسيقات المهيكلة الأخرى.

البيانات الضخمة هي في الأساس تطبيق خاص لعلم البيانات حيث تكون مجموعة البيانات كبيرة جداً وتحتاج إلى التغلب على التحديات المنطقية لمواجهتها. علم البيانات هو نهج علمي يطبق الأفكار الخوارزمية والحاسوبية والحوسبية لمعالجة هذه البيانات الضخمة.

بشكل عام ، الشاغل الرئيسي للبيانات الضخمة هو التخزين والاستخراج والمعالجة والتحليل في هذه المجموعة الضخمة من البيانات. غالبًا ما يتعذر معالجة وتحليل مجموعة البيانات الكبيرة

هذه بسبب القيود الحسابية. لذلك ، يتطلب أساليب وأدوات خاصة ، مثل الخوارزميات والبرمجيات والبرمجة المتوازية ، إلخ.

الفرق بين علم البيانات والبيانات الضخمة

المدرجة أدناه هي الاختلافات بين علم البيانات والبيانات الضخمة:

- تحتاج المؤسسات إلى البيانات الضخمة لتحسين أدائها وزيادة نمو أعمالها وتقديم منتجات أفضل لعملائها. بينما يوفر علم البيانات أساليب وآليات لفهم واستغلال إمكانات البيانات الضخمة في الوقت المناسب.
- في الوقت الحاضر ، لا يوجد حد لكمية البيانات القيمة التي يمكن جمعها للمنظمات. لكن علم البيانات ضروري لاستخدام كل هذه البيانات لاستخراج معلومات ذات مغزى لاتخاذ قرارات تنظيمية.
- من الواضح أن علم البيانات يستخدم مناهج نظرية وعملية لاستكشاف معلومات البيانات الضخمة، والتي تلعب دورًا مهمًا في استغلال إمكانات البيانات الضخمة. يمكن اعتبار البيانات الضخمة على أنها مجموعة من البيانات غير الصالحة ، إلا إذا تم تحليلها بالاستدلال الاستنباطي والاستقرائي.
- يرتبط تحليل البيانات الضخمة بالتنقيب في البيانات. لكن علم البيانات يستخدم خوارزميات التعلم الآلي لتصميم وتطوير النماذج الإحصائية لتوليد المعرفة بكميات كبيرة من البيانات الضخمة.

من الاختلافات المذكورة أعلاه بين البيانات الضخمة وعلم البيانات ، قد يكون من الواضح أن علم البيانات مدرج في مفهوم البيانات الضخمة. يلعب علم البيانات دورًا مهمًا في العديد من المجالات التطبيقية. يعمل علم البيانات على البيانات الضخمة للحصول على نتائج مفيدة من خلال التحليل التنبئي ، حيث يتم استخدام النتائج لاتخاذ قرارات ذكية.

يمكن فهم الفرق الأساسي بين البيانات الضخمة وعلم البيانات بشكل أفضل من خلال إعطاء مثال. علم البيانات مثل كتاب يمكنك من خلاله إيجاد حل لمشاكلك. من ناحية أخرى ، يمكن اعتبار البيانات الضخمة مكتبة كبيرة حيث توجد جميع الإجابات على أسئلتك ، ولكن من الصعب العثور على إجابات لأسئلتك.

لماذا نتعلم علم البيانات؟

نحن نعيش في أكثر الأوقات إثارة في تاريخ البشرية. الفترة التي أصبحت فيها البيانات سلعة أكثر قيمة من النفط والذهب. إن كمية البيانات التي يتم إنتاجها عالميًا غير مسبوقة ومن المتوقع أن تستمر مع نمو سكان العالم وزيادة إمكانية الوصول إلى الإنترنت.

هذه البيانات التي تم إنشاؤها هي مصدر قيم للغاية وهي علم في فك تشفيرها. أيضًا، كانت هناك تغييرات واسعة النطاق في نمط سلوك العملاء في التسوق عبر الإنترنت، والاستثمار في سوق

الأوراق المالية، وما إلى ذلك، وكل منها يتطلب الحاجة إلى تحليل عميق للبيانات. هذا هو المكان الذي يلعب فيه علم البيانات ويطلبه الشركات والمؤسسات وما إلى ذلك.

تحليل البيانات

تحليل البيانات هو عملية جمع ونمذجة وتحليل البيانات لاستخراج الأفكار المفيدةفي صنع القرار. يُطلق على الشخص المسؤول عن هذه التحليلات محلل البيانات. يستخرج محلل البيانات البيانات من خلال عدة طرق، مثل تنظيف البيانات وتحويل البيانات ونمذجة البيانات. هناك عدة طرق وأساليب للتحليل، حسب الصناعة والغرض. يسمح تحليل البيانات للصناعات بمعالجة الاستفسارات بسرعة وتحقيق نتائج عملية مطلوبةفي فترة زمنية قصيرة.

يمكن لمحللى البيانات، بغض النظر عن الصناعة التى يعملون فيها، قضاء وقتهم فى تطوير أنظمة لجمع البيانات وجمع نتائجهم فى تقارير يمكن أن تساعد شركاتهم على النمو والتحسين.

يمكن أن يشارك محللو البيانات فى أى جزء من عملية التحليل. بصفتك محلل بيانات، يمكنك المشاركة فى كل شىء بدءًا من إعداد نظام تحليلات إلى تقديم رؤى بناءً على البيانات التى تجمعها. قد يُطلب منك تدريب الآخرين فى نظام جمع البيانات الخاص بك.

على الرغم من أن علم البيانات وتحليل البيانات قد يأتي من مجال إحصائي مشترك ، إلا أن خرائطهم وجذورهم مختلفة. ومع ذلك ، يعتقد معظم الناس أن علم البيانات وتحليل البيانات متشابهان. من أجل فهم اختلافاتهم ، يجب علينا تقييمهم. بعض هذه الاختلافات مذكورة أدناه:

- يتمثل الاختلاف الرئيسي الأول بين عالم البيانات ومحلل البيانات في أنه بينما يتعامل محلل البيانات مع حل المشكلات ، يحدد عالم البيانات المشكلات ثم يحلها. يتم توظيف محللي البيانات من قبل الشركات لحل مشاكلهم التجارية. يتمثل دور محلل البيانات في العثور على اتجاهات مبيعات أفضل أو استخدام إحصاءات موجزة لوصف معاملات العملاء. من ناحية أخرى ، لا يحل عالم البيانات المشكلات والقضايا فحسب ، بل يحدد أيضًا المشكلات في المقام الأول.
- لا يحتاج محللو البيانات إلى مهارات الاتصال أو الفطنة التجارية. محلل البيانات يقتصر على حدود تحليل البيانات. ليست هناك حاجة لإيصال النتائج إلى الفريق ومساعدتهم على اتخاذ قرارات قائمة على البيانات. ومع ذلك ، يجب أن يكون لدى عالم البيانات مهارات في سرد القصص ومهارات الإدارة من أجل ترجمة النتائج التي توصل إليها إلى استراتيجيات الأعمال. لذلك ، يلعب عالم البيانات دورًا مهمًا في عملية صنع القرار في الشركة.
- هناك اختلاف آخر بين عالم البيانات ومحلل البيانات وهو الاختلاف في معالجة البيانات. يستخدم محللو البيانات استعلامات SQL لاسترداد البيانات المنظمة وإدارتها. في المقابل،

يستخدم علماء البيانات NoSQL للبيانات غير المهيكلة. لذلك ، فإن علماء البيانات مسؤولون عن إدارة كل من أنواع البيانات غير المهيكلة والمنظمة.

- لا يتعامل محلل البيانات مع تطوير النمذجة التنبؤية أو الأدوات الإحصائية للتنبؤ بالبيانات. ومع ذلك ، يحتاج علماء البيانات إلى معرفة التعلم الآلي لبناء نماذج تنبؤ قوية. نماذج التنبؤ هذه هي نماذج التوقع والتصنيف.
- يحتاج علماء البيانات إلى تعديل نماذج البيانات لإنشاء منتجات بيانات بشكل أفضل. يتطلب أيضًا تحسين أداء نماذج التعلم الآلي. هذا غير مطلوب من قبل محللي البيانات. وبالتالي، لا يقتصر دور عالم البيانات على بناء النماذج فحسب، بل يشمل أيضًا إعدادها وصيانتها.

يتداخل تحليل البيانات وعلوم البيانات بشكل كبير مع بعضهما البعض وفى نفس الوقت يختلفان تمامًا عن بعضهما البعض. يكون تركيز محللو البيانات انياً ، بينما يتنبأ علماء البيانات بما قد يحدث.

غالبًا ما يكون محللو البيانات علماء بيانات ذوى مستوى منخفض يقضون معظم وقتهم فى تحليل البيانات وتقديم الاقتراحات. ومع ذلك ، فهم عادة لا يحتاجون إلى إنشاء برامج تقنية وخوارزميات التعلم الآلى. هذا لأن محللى البيانات ، على عكس علماء البيانات ، ليس لديهم علاقة كبيرة بالتحليلات التنبؤية. إنهم يعملون مع البيانات الموجودة ويقدمون ملخصًا لمجموعة متنوعة من تفاصيل أداء الشركة.

المسؤوليات الرئيسية لمحلل البيانات

الإجابة على السؤال "**ماذا يفعل محلل البيانات؟**" سيختلف اعتمادًا على المنظمة ومدى اتخاذ القرارات المستندة إلى البيانات من قبل الشركة. ومع ذلك ، فإن مسؤوليات محلل البيانات تشمل عادة ما يلي:

- **تطوير وتنفيذ قواعد البيانات وأنظمة جمع البيانات والاستراتيجيات الأخرى التي تعمل على تحسين الكفاءة والجودة الإحصائية.**
- **استخراج البيانات من المصادر الأولية والثانوية ، ثم أعادة تنظيم البيانات في تنسيق يمكن قراءته بسهولة بواسطة البشر أو الآلات.**
- **استخدام الأدوات الإحصائية لتفسير مجموعات البيانات ، مع إيلاء اهتمام خاص للاتجاهات[1] والأنماط التي يمكن أن تكون ذات قيمة في التشخيص والتحليل التنبئي.**
- **التعاون مع المبرمجين والمهندسين والقادة التنظيميين لتحديد فرص تحسين العملية واقتراح تعديلات على النظام وتطوير سياسات إدارة البيانات.**

---

[1] Trends

- **تقديم الوثائق المناسبة التي تسمح لأصحاب المصلحة بفهم خطوات عملية تحليل البيانات وتكرار التحليل إذا لزم الأمر.**

أنواع تحليل البيانات

يعد تحليل البيانات أداة أساسية لتحسين الأداء العام لأي شركة أو مؤسسة. من خلال تنفيذ بعض تحليلات البيانات في نماذج أعمالها ، من المرجح أن تكون الشركات قادرة على اتخاذ قرارات أفضل وتحسين عملياتها وتقليل تكاليفها. هناك أربعة أنواع من تحليل البيانات تنطبق على جميع الصناعات. ولكن ما هو أفضل نوع من نماذج تحليل البيانات لأي شركة أو مؤسسة؟

بينما نقسم هذه الأقسام إلى فئات منفصلة ، فجميعها مترابطة ومتصلة ببعضها البعض. تنقسم طرق التحليل إلى أربع فئات رئيسية حسب درجة التعقيد: التحليل الوصفي والتشخيصي والتنبؤي والتوجيهي. كلما انتقلت من الأبسط إلى الأكثر تعقيدًا ، تزداد الصعوبة والموارد المطلوبة ، وكذلك مستوى البصيرة.

التحليل الوصفي [1]- ماذا حدث

التحليل الوصفي هو الخطوة الأولى والأكثر شيوعًا والأبسط في إجراء أي عملية تحليل إحصائي، والغرض منه هو الإجابة عن السؤال ، ماذا حدث؟ بمعنى آخر ، فإنه يفحص وصفًا لما حدث في الماضي: الإيرادات الشهرية ، ومبيعات الأشهر الستة الماضية ، والزيارات السنوية لموقع الويب، وما إلى ذلك. يتم ذلك عن طريق تلخيص البيانات السابقة وبناء وتفسير البيانات الأولية من مصادر مختلفة لتحويلها إلى رؤى قيمة. يوضح لنا هذا التحليل كيفية توزيع البيانات ، ويساعد في تحديد المكبات ، ويمكننا من تحديد العلاقة بين المتغيرات ، وبالتالي إعداد البيانات لمزيد من التحليل الإحصائي.

كيف يمكن أن يساعد التحليل الوصفي في العالم الحقيقي؟ على سبيل المثال ، في أحد أقسام الرعاية الصحية ، ينص على أنه يتم إدخال عدد كبير من الأشخاص إلى غرفة الطوارئ في فترة زمنية قصيرة. يخبرك التحليل الوصفي أن هذا يحدث ويوفر بيانات فورية[2] مع جميع الإحصائيات ذات الصلة (تاريخ الحدوث ، والحجم ، وتفاصيل المريض ، وما إلى ذلك).

التحليل الوصفي ضروري، لأنه يسمح لنا بتقديم بياناتنا بطريقة هادفة. ومع ذلك، تجدر الإشارة إلى أن هذا التحليل وحده لن يسمح لك بالتنبؤ بالنتائج المستقبلية أو الإجابة على أسئلة مثل سبب حدوث شيء ما، ولكن يتم تنظيم بياناتك وتحويلها بواسطة مجموعات البيانات. فهي جاهزة لإجراء المزيد من التحليل. وعادة ما يقوم بذلك عن طريق تلخيص وإبراز الميزات

---

[1] Descriptive Analysis

[2] Real-time

الرئيسية المفضلة لدينا بالإضافة إلى استخدام المخططات وغيرها من العروض سهلة الاستخدام.
الجانب السلبي هو أن التحليل الوصفي يسلط الضوء على ظاهرة دون أن يشرح بالضبط سبب حدوثها. لهذا السبب، يجب دائمًا دمج هذا النهج مع التحليلات الأخرى للحصول على الفوائد الحقيقية لعملك.

التحليل التشخيصي[1] - لماذا حدث هذا

التحليل التشخيصي، الذي يشار إليه غالبًا باسم تحليل السبب الجذري[2] ، هو نوع متقدم من التحليل يتقدم خطوة أخرى إلى البيانات أو المحتوى للإجابة على السؤال "لماذا حدث هذا؟". يتميز التحليل التشخيصي بأساليب مثل التنقيب عن البيانات والارتباط، ويلقي نظرة أعمق على البيانات لفهم أسباب الأحداث والسلوكيات، مما يسمح لك بفهم معلوماتك بسرعة أكبر والإجابة على الأسئلة المهمة من القوى العاملة.في مثال الرعاية الصحية المذكور سابقًا، يفحص التحليل التشخيصي البيانات ويجعل العلاقات المتبادلة. على سبيل المثال، قد يساعدفي تحديد ما إذا كانت جميع أعراض المريض: "ارتفاعفي درجة الحرارة، وسعال جاف، وإرهاق" تشير إلى عامل معدي. الآن لديك تفسير للزيادة المفاجئةفي الحجمفي غرفة الطوارئ.

مفتاح نجاح هذا النهج هو الوصول إلى البيانات على نطاق واسع. مثلا التحليل الوصفي، يتطلب التحليل التشخيصي بيانات "داخلية" سابقة، ولكن بخلاف التحليل السابق، غالبًا ما يتضمن التحليل التشخيصي معلومات خارجية من مجموعة واسعة من المصادر لتحديد ما حدث بالتفصيل. على سبيل المثال، قد تجد أن إيرادات موقع الويب الخاص بك قد انخفضتفي الربع الأخير. يمكن أن يكون هذا متعلقًا بخفض تكاليف الإعلان بالإضافة إلى التغييراتفي خوارزمية Google. للعثور على أدلة لهذه العملية، من الضروري تحديد البيانات من مصادر مختلفة مثل السجلات الجديدة وغيرها للعثور على نمط مشبوه.

التحليل التنبئي [3]- ماذا سيحدث

يحدد التحليل التنبئي النتائج المحتملة من خلال تحديد الميولفي التحليلات التشخيصية والوصفية. يأخذ التحليل التنبئي البيانات السابقة ويغديهافي نموذج التعلم الآلي الذي يأخذفي الاعتبار الأنماط الرئيسية. ثم يتم تطبيق النموذج على البيانات الحالية للتنبؤ بما سيحدث. هذا يسمح للمنظمة لاتخاذ إجراءات وقائية. على سبيل المثال، مثل الاتصال بعميل من غير المرجح أن يجدد عقده.في المستشفى التي ذكرناها، قد يتنبأ التحليل التنبئي بزيادة عدد المرضى الداخليين

---

[1] Descriptive Analysis
[2] Root cause analysis
[3] Predictive Analysis

في قسم الطوارئ خلال الأسابيع القليلة القادمة. بناءً على الأنماط الموجودةفي البيانات، ينتشر المرض بسرعة.

التحليل التوجيهي[1]- ما هي الإجراءات التي يجب اتخاذها

التحليل التوجيهي هي عملية تحلل البيانات وتقدم نصائح فورية حول كيفية تحسين ممارسات الأعمال لتناسب العديد من النتائج المتوقعة.في الواقع، التحليل الوصفي "يأخذ ما نعرفه (البيانات) "، ويفهم البيانات للتنبؤ بما يحدث، ويقترح أفضل الخطوات التطلعية بناءً على المحاكاة الواعية، ويحدد العواقب المحتملة لكل منها. الغرض من التحليل التوجيهي، وهو بالتأكيد المجموعة الأكثر تقدمًافي قائمتنا، هو اقتراح طريقة عملية لتجنب المشاكل المستقبلية أو لتحقيق أقصى استفادة من عملية واعدة.

يرتبط التحليل التوجيهي بالتحليل الوصفي والتنبؤي. بينما يسعى التحليل الوصفي إلى نمذجة والتنبؤ بما قد يحدث من خلال توفير نظرة ثاقبة لما حدث والتحليل التنبئي، يسعى التحليل التوجيهي إلى تحديد الأفضل، وفقًا للمعايير المعروفة. يكون الحل أو النتيجة من بين الخيارات المختلفة.

يمكن أن يقترح التحليل التوجيهي أيضًا خيارات اتخاذ القرار بشأن كيفية الاستفادة من فرصة مستقبلية أو تقليل المخاطر المستقبلية، وإظهار عواقب كل خيار من خيارات اتخاذ القرار.في الممارسة العملية، يمكن للتحليل التوجيهي معالجة البيانات الجديدة بشكل مستمر وتلقائي لتحسين دقة التنبؤ وتوفير خيارات أفضل لصنع القرار. دعنا نعود إلى المستشفى، على سبيل المثال: الآن بعد أن عرفت أن المرض ينتشر، قد تشير أدوات التحليل التوجيهي إلى زيادة عدد الموظفين من أجل العلاج الفعال للمرضى.

التحليل التوجيهي هو تطور طبيعي لأساليب التحليل الوصفي والتنبؤي، ويذهب خطوة أخرى إلى الأمام لإزالة التخمين[2] من تحليل البيانات. كما أنه يوفر الوقت لعلماء البيانات والمسوقين الذين يحاولون فهم ما تعنيه بياناتهم وأين يمكن توصيلها لتوفير تجربة مستخدم مخصصة بالكامل.
تستخدم المنظمات المستقبلية مجموعة متنوعة من التحليلات لاتخاذ قرارات ذكية تساعد الشركات أو تنقذ الأرواح في المستشفى المذكورة.

---

[1] Prescriptive Analysis

[2] Guesswork

دورة حياة[1] علم البيانات

تتكون دورة حياة علم البيانات من خمس مراحل. علماء البيانات الفعالون هم أولئك الذين يمكنهم أداء أي من هذه المراحل. هذه الخطوات الخمس هي على النحو التالي:

- **المرحلة الاولى: جمع البيانات**

كما يوحي الاسم، هذا هو المكان الذي يتم فيه جمع البيانات. ينشئ المستخدمون ملايين البيانات كل يوم. يتم إضافة كل رابط يتم النقر عليه وإجراء بحث وتحميل صورة وإرسال رسالة إلى مستودع البيانات. على هذا النحو، فإن عملية جمع البيانات ليست سهلة. يجب تحديد البيانات ذات الصلة بالمشروع. هناك أيضًا مهمة تحديد موقع جمع البيانات. تتوفر أنواع مختلفة من مصادر البيانات عبر الإنترنت من خلال المواقع الإخبارية واستطلاعات الرأي وما إلى ذلك على الموقع ووسائل التواصل الاجتماعي.

- **المرحلة الثانية: تنظيف البيانات**

من المهم معرفة أن البيانات التي تم جمعهافي المرحلة الأولى غير منظمة. يجب على عالم البيانات تنقية البيانات الخام وتصنيفها. هذا يعني البحث عن أي تناقضات (بيانات مكررة، بيانات شاذة، إلخ)في البيانات من أجل تجنب أي أخطاءفي الخطوات التالية. بسبب المهام التي يتم تنفيذها، عادة ما تكون المرحلة الثانية هي الجزء الذي يستغرق وقتًا طويلاًفي مشروع علمي.

- **المرحلة الثالثة: تنقيب البيانات**

يبدأ التحليل بعد تنظيف مجموعة البيانات. يقوم علماء البيانات بفحص البيانات للحصول على فكرة أوسع عن الأنماط والاتجاهات الرئيسيةفي مجموعة البيانات. يسيطر التمثيل المرئي والتحليل الإحصائي على هذه المرحلة. يسلط تنقيب البيانات الضوء على النقاط التي تحتاج إلى مزيد من التحليل. تسمح أدوات التمثيل المرئي للبيانات أيضًا لعلماء البيانات بتسجيل واستكشاف العناصر المهملة بشكل أكبر.

- **المرحلة الرابعة: نمذجة البيانات**

النمذجة هيفي صميم أساليب البحثفي علوم البيانات. إنه فهم العلاقة بين عناصر البيانات وتخطيطها. من خلال نمذجة البيانات، يرى عالم البيانات كيف تتفاعل العناصر الأكثر أهمية وتتجمع معًا. هناك عدة طرق لبناء نموذج. يمكن أن تكون هذه النمذجة من خلال

[1] Life Cycle

أساليب التعلم الآلي أو النمذجة الإحصائية. فقط بعد النمذجة يبدأ عالم البياناتفي استخلاص الرؤى منها.

- **المرحلة الخامسة: تفسير البيانات**

بمجرد استخلاص رؤى ذات مغزى من البيانات، فقد حان الوقت للخطوة الأخيرةفي دورة حياة علم البيانات، وهي تفسير البيانات. إذا كنت تريد تنفيذ اكتشافاتك القيمة، فيجب أن تكون قادرًا على تقديم هذه الأفكار بطريقة جذابة ومفهومة حتى يتمكن أصحاب المصلحة في المشروع من فهمها بسهولة.

عالم البيانات

علماء البيانات هم جيل جديد من محللي البيانات الذين لديهم المهارات التقنية لحل المشكلات المعقدة ولديهم فضول أيضًا لاكتشاف المشكلات التي تحتاج إلى حل. تتمثل مهمة عالم البيانات في عمل التنبؤات الأكثر دقة. وهذا يتطلب استخدام التقنيات التحليلية المتقدمة ، بما في ذلك التعلم الآلي والنمذجة التنبؤية. دور عالم البيانات هو مزيج من عدة أدوار فنية تقليدية ، بما في ذلك عالم الرياضيات والإحصائي وأخصائي الكمبيوتر. لم يحظ علماء البيانات بالكثير من الاهتمام قبل عقد من الزمن ، لكن شعبيتهم المفاجئة تعكس كيف تفكر الشركات الآن في البيانات الضخمة. لم يعد بالإمكان تجاهل الكثير من البيانات غير المهيكلة ونسيانها. هذا منجم ذهب افتراضي يساعد على زيادة الإيرادات.

بدأ العديد من علماء البيانات حياتهم المهنية كإحصائيين أو محللي بيانات. ولكن مع بدء تطور البيانات الضخمة (وتخزين البيانات الضخمة وتقنيات المعالجة) ، بدأت هذه الادوار في التطور. لم تعد البيانات مجرد خلفية لمعالجة تكنولوجيا المعلومات. هذه معلومات أساسية تتطلب التحليل والفضول الإبداعي ومهارة لترجمة الأفكار التكنولوجية المتطورة إلى طرق جديدة للاستفادة.

في مجال الأعمال ، يعمل علماء البيانات عادةً في فرق لاستخدام البيانات الضخمة للمعلومات التي يمكن استخدامها للتنبؤ بسلوك العملاء وتحديد فرص الاستثمار الجديدة. في العديد من المؤسسات ، يتم تكليف علماء البيانات أيضًا بتحديد أفضل الطرق لجمع البيانات ، باستخدام أدوات لتحليل البيانات وتفسيرها.

يدير علماء البيانات دورة حياة البيانات بأكملها، من التجميع والتنظيم إلى التحليل والتفسير. رؤيتهم عادة ما تكون استشرافية. وهذا يعنى أنهم يقومون بتقييم البيانات السابقة ذات الصلة واستخراج الأفكار التى يمكن استخدامها كأساس للتغييرات المحتملة فى سلوك المستهلك أو الاتجاهات. هذا يسمح للمنظمات بوضع استراتيجيات طويلة المدى.

مهندس البيانات

يكون عالم البيانات الجيد جيداً مثل البيانات التي يمكنه الوصول إليها. تقوم معظم الشركات بتخزين بياناتها بتنسيقات مختلفةفي قواعد بيانات وملفات نصية. هذا هو المكان الذي يأتي فيه مهندسو البيانات؛ يقومون ببناء خط أنابيب يحول هذه البيانات إلى تنسيقات يمكن لعلماء البيانات استخدامها. مهمتهم الرئيسية هي توفير البيانات للاستخدامات التحليلية أو التشغيلية. يقومون بدمج البيانات وتحليلها واستخدامهافي تطبيقات التحليلات. هدفهم هو تسهيل الوصول إلى البيانات وتحسين الايكوسيستم للبيانات الضخمة لمؤسستهم. مهندسو البيانات لا يقلون أهمية عن علماء البيانات، لكنهم عادة ما يكونون أقل وضوحًا.

بينما يستكشف علم البيانات وعلماء البيانات البيانات على وجه التحديد، وإيجاد رؤى بشأنها، وبناء خوارزميات التعلم الآلى، يهتم مهندس البيانات بوظيفة هذه الخوارزميات فى البنية التحتية للإنتاج والبنية التحتية لخطوط البيانات. وبالتالى، فإن مهندس البيانات له دور هندسى فى فريق علوم البيانات أو أى مشروع متعلق بالبيانات، حيث يحتاج إلى إنشاء وإدارة البنية التحتية التكنولوجية لمنصة البيانات.

دور مهندس البيانات

يركز مهندسو البيانات على جمع البيانات وإعدادها لاستخدامها من قبل علماء البيانات والمحللين. يأخذ مهندسو البيانات ثلاثة أدوار رئيسية على النحو التالي:

- **خبير عام:** عادة ما يعمل مهندسو البيانات المركّزون بشكل عام في فرق صغيرة. قد يكونون أكثر مهارة من معظم مهندسي البيانات ، لكن لديهم معرفة أقل بهندسة النظم. يتناسب عالم البيانات الذي يريد أن يكون مهندسًا بشكل جيد مع دور الخبير العام. بدون مهندس بيانات ، ليس لدى المحللين وعلماء البيانات ما يحللونه. ومن ثم ، فإن مهندس البيانات هو عضو مهم في فريق علم البيانات.
- **خط أنابيب_المحور:** في الشركات المتوسطة الحجم ، غالبًا ما يعمل مهندسو البيانات هؤلاء جنبًا إلى جنب مع علماء البيانات لتحقيق أقصى استفادة من البيانات التي يتم جمعها. تعد المعرفة بعلوم الكمبيوتر والأنظمة الموزعة أمرًا ضروريًا للمهندسين المعتمدين على خطوط الأنابيب لإجراء مثل هذه التحليلات.
- **قاعدة البيانات_المحور:** مهندسو البيانات هؤلاء مسؤولون عن تنفيذ قواعد بيانات التحليلات وصيانتها وتجميعها. يوجد هذا الدور عادةً في الشركات الكبرى حيث يتم توزيع البيانات عبر قواعد بيانات متعددة. إدارة تدفق البيانات هي وظيفة بدوام كامل ، ويركز مهندسو البيانات في هذا الدور بالكامل على تحليلات قاعدة البيانات. بصفتك مهندسًا يعتمد على

قاعدة البيانات ، يجب أن تعمل على قواعد بيانات متعددة وأن تطور جداول بيانات في قاعدة بيانات.

المجالات والمهارات الأساسية للدراسة فى علم البيانات

علم البيانات هو مصطلح واسع يتطلب الكفاءة في مختلف المجالات لإتقانها. المدرجة أدناه هي بعض المجالات والجوانب الرئيسية اللازمة لإتقان علم البيانات:

تعلم الآلة

بالنسبة لعالم البيانات ، يعد تعلم الآلة مهارة أساسية. تتمثل الفكرة الأساسية للتعلم الآلي في السماح للآلات بالتعلم بشكل مستقل باستخدام كتلة البيانات التي يتم تغذيتها بالجهاز كمدخلات. مع تقدم التكنولوجيا ، يتم تدريب الآلات على التصرف مثل البشر في القدرة على اتخاذ القرار.

التعلم العميق

غالبًا ما يستخدم التعلم العميق في علم البيانات. لأنها تعمل بشكل أفضل بكثير من طرق التعلم الآلي التقليدية. بالمقارنة مع التعلم الآلي مع التعلم العميق ، يمكن القول أنه بينما يستخرج التعلم العميق الميزات تلقائيًا من بنية البيانات ، يجب أن يتم ذلك يدويًا عن طريق التعلم الآلي ، وإذا كان يقوم بتنبؤات غير صحيحة في عملية صنع القرار ، فإن الخبير أو المبرمج يجب أن تتناول هذه المشكلة صراحة.

الرياضيات

لتحسين مهارات التعلم الآلي، يجب أن يكون لدى عالم البيانات معرفة عميقة بالرياضيات. موضوعان مهمان في الرياضيات من حيث التطبيق في علم البيانات هما الجبر والحساب. في حين أن الجبر يدور حول دراسة المتجهات والوظائف الخطية ، فإن الحساب هو الدراسة الرياضية للتغير المستمر. يتم استخدام العديد من المفاهيم الجبرية مثل الموترات والمتجهات في العديد من مجالات التعلم الآلي. وبالمثل ، فإن الحسابات مطلوبة في مجالات مختلفة من التعلم الآلي، مثل تقنيات التحسين.

الاحصاء والاحتمالية

العالم هو عالم احتمالي ، لذلك نحن نعمل مع البيانات الاحتمالية ؛ هذا يعني أنه وفقًا لمجموعة محددة من المتطلبات الأساسية ، ستظهر لك البيانات جزءًا من الوقت فقط. لاستخدام علم البيانات بشكل صحيح ، يجب أن يكون المرء على دراية بالاحتمالات والإحصاءات. الإحصاء والاحتمالات من المتطلبات الأساسية في علم البيانات والمعرفة الجيدة في هذا المجال ضرورية.

معالجة اللغة الطبيعية

في مجال علم البيانات ، تعد معالجة اللغة الطبيعية مكونًا مهمًا للغاية مع تطبيقات واسعة في مختلف قطاعات الصناعة والشركات. من السهل على البشر فهم اللغة ، ومع ذلك ، فإن الآلات غير قادرة على التعرف عليها بشكل كافٍ. معالجة اللغة الطبيعية هي فرع من فروع الذكاء الاصطناعي يركز على سد الفجوة بين التواصل بين الإنسان والآلة لتمكين الآلة من التفسير والفهم.

العرض المرئى للبيانات

يعد تصوير البيانات أحد أهم فروع علم البيانات. ببساطة ، يتضمن الرسم التوضيحي عرض البيانات في شكل رسوم بيانية ومخططات بيانية.

لغة البرمجة

يجب أن يتمتع عالم البيانات ، بالإضافة إلى مهارات الكمبيوتر الأساسية مثل الكفاءة في Microsoft Excel ، بمهارات البرمجة حتى يتمكن من استخدامها للعمل مع البيانات (المعالجة والتمثيل المرئي وما إلى ذلك) واستخدام التعلم الآلي ومهارات التعلم العميق في تنفيذ المشروع.

الخوارزميات

نظرًا لأن جميع أنظمة التعلم الآلي تعتمد على الخوارزميات ، فمن الضروري جداً أن يكون لدى عالم البيانات فهم أساسي للخوارزميات وكيفية تصميمها.

تطبيقات علم البيانات

الآن بعد أن عرفت أهمية علم البيانات والمتطلبات الأساسية والمهارات اللازمة له ، من المهم أن تعرف كيف يمكن استخدام علم البيانات في العالم الحقيقي ، وسنرى كيف غيّر علم البيانات العالم اليوم. لذلك ، إليك قائمة بتطبيقات علوم البيانات لمعرفة المزيد عن تطبيقاتها:

المواصلات

أهم تقدم أو تطوير حققه علم البيانات في مجال النقل هو إدخال السيارات ذاتية القيادة. لقد أسس علم البيانات موطئ قدم قوي في صناعة النقل من خلال التحليل المكثف لأنماط استهلاك الوقود ، والمراقبة النشطة للمركبة وسلوك السائق ، وتوفير بيئات قيادة أكثر أمانًا للسائقين، وتحسين أداء السيارة ، وإضافة الاستقلالية إلى السيارات وغير ذلك. باستخدام التعلم المعزز والاستقلالية ، يمكن لشركات صناعة السيارات بناء سيارات أكثر ذكاءً وطرقًا منطقية أفضل.

كشف المخاطر والاحتيال

تم استخدام علم البيانات لأول مرة في التمويل والمصارف. كانت العديد من المؤسسات المالية مثقلة بالديون في نهاية كل عام. لذلك ، تم اعتبار اساليب علم البيانات كحل. لتحليل احتمالية المخاطر ، تعلموا فصل البيانات بناءً على مواصفات العميل والتكاليف السابقة والمتغيرات الضرورية الأخرى. وبالتالي ، يمكنهم القيام بالتسويق المستهدف بناءً على إيرادات كل عميل كل عام.

علم الوراثة والجينات الوراثية

يساعد علم البيانات علماء الأحياء على تحليل استجابة الجينات للأدوية المختلفة. والغرض منه هو فهم ودراسة تأثير الحمض النووي على صحة الشخص، والذي يسعى إلى إيجاد روابط بيولوجية بين الأمراض والجينات والاستجابات للأدوية.

تطوير الادوية

يتطلب اكتشاف دواء جديد سنوات من البحث والاختبار للوصول إلى مرحلة الإنتاج وفي النهاية يتم ترخيصه للمتاجر الطبية والمستشفيات للمرضى. يمكن استخدام خوارزميات التعلم الآلي وعلوم البيانات لتبسيط العملية وتقليل الوقت اللازم للفحص الأولي لمركبات الأدوية المستخدمة في إنتاج الأدوية. يمكن أن تتنبأ الخوارزميات وعلوم البيانات أيضًا بكيفية استجابة الجسم لمركبات دوائية معينة باستخدام نماذج ومحاكاة إحصائية ورياضية مختلفة. هذا أسرع بكثير من الاختبارات المختبرية التقليدية. يمكن للنماذج أيضًا توقع النتائج المستقبلية بشكل أكثر دقة.

خلاصة الفصل الاول

- أصبحت الصناعات تعتمد على البيانات وتحتاج إلى علماء بيانات لمساعدتها على اتخاذ قرارات أكثر ذكاءً وإنشاء منتجات أفضل.
- الغرض من علم البيانات هو شرح العمليات من خلال البيانات الموجودة.
- تشير البيانات الضخمة إلى مجموعة كبيرة من البيانات غير المتجانسة التي تم الحصول عليها من مجموعة متنوعة من المصادر.
- البيانات الضخمة هي في الأساس تطبيق خاص لعلم البيانات.
- تنقسم طرق التحليل إلى أربع فئات رئيسية حسب درجة التعقيد: التحليل الوصفي والتشخيصي والتنبؤي والتوجيهي.
- تتمثل مهمة عالم البيانات في عمل التنبؤات الأكثر دقة.
- يكون عالم البيانات الجيد جيداً مثل البيانات التي يمكنه الوصول إليها.

- يركز مهندسو البيانات على جمع البيانات وإعدادها لاستخدامها من قبل العلماء ومحللي البيانات.

مصادر اضافية لمزيد من القراءة

Blum, A., Hopcroft, J., & Kannan, R. (2020). Foundations of data science. Cambridge University Press.

Godsey, B. (2017). Think Like a Data Scientist: Tackle the data science process step-by-step. Simon and Schuster.

Kotu, V., & Deshpande, B. (2018). Data science: concepts and practice. Morgan Kaufmann.

Pearson, R. K. (2018). Exploratory data analysis using R. CRC Press.

Peng, R. D., & Matsui, E. (2016). The Art of Data Science: A guide for anyone who works with Data. Skybrude consulting LLC.

Pierson, L. (2021). Data science for dummies. John Wiley & Sons.

Yu, K. (2019). Data Science Foundations: Geometry and Topology of Complex Hierarchic Systems and Big Data Analytics. Boca Raton, FL: Chapman & Hall/CRC Press.

O'Neil, C., & Schutt, R. (2013). Doing data science: Straight talk from the frontline. " O'Reilly Media, Inc.".

Sharda, R., Delen, D., & Turban, E. (2020). Analytics, data science, & artificial intelligence: Systems for decision support. Pearson Education, Inc..

Bruce, P., & Bruce, A. (2017). Practical statistics for data scientists: 50 essential concepts. " O'Reilly Media, Inc.".

Dangeti, P. (2017). Statistics for machine learning. Packt Publishing Ltd.

Irizarry, R. A. (2019). Introduction to data science: data analysis and prediction algorithms with R. CRC Press.

Nandi, G., & Sharma, R. K. (2020). Data Science Fundamentals and Practical Approaches: Understand Why Data Science Is the Next. BPB Publications.

Kroese, D. P., Botev, Z. I., Taimre, T., & Vaisman, R. (2019). Data science and machine learning: mathematical and statistical methods. Chapman and Hall/CRC.

2

مقدمة في بايثون

الأهداف:

- سبب اختيار بايثون في علم البيانات.
- مفاهيم البرمجة في بايثون.
- مقدمة في مكتبة NumPy

ما هي لغة بايثون؟

كعالم بيانات جديد، يبدأ مسارك بلغة البرمجة التي تحتاج إلى تعلمها. من بين جميع اللغات التي يمكنك الاختيار من بينها، فإن بايثون Python هي الأكثر شعبية بالنسبة للعالم. بايثون هي لغة برمجة عالية المستوى موجهة للكائنات وتستخدم لمجموعة واسعة من المشكلات ذات النطاق والتعقيد المتفاوتين. على عكس العديد من اللغات المماثلة، من السهل إتقانها وتعلمها وهي مثالية للمبتدئين. لكن هذه السهولة ليست السبب الوحيد لأهميتها، هي قوية بما يكفي حتى للمستخدمين المتقدمين. بالإضافة إلى ذلك، تعد بايثون أكثر أدوات علوم البيانات استخدامًا، وهي مدرجة كشرط في معظم قوائم إعلانات وظائف علوم البيانات.

سبب اختيار بايثون في علم البيانات؟

تعد بايثون واحدة من أكثر اللغات قيمة وإثارة للاهتمام لتحليل البيانات وتتزايد شعبيتها في عالم تحليل البيانات وعلوم البيانات يومًا بعد يوم. نظرًا لأن بايثون هي واحدة من أكثر لغات البرمجة مرونة، فقد أصبحت علمًا مفضلًا. أيضًا، يفضل الأشخاص الذين يرغبون في دخول عالم العلوم لغة بايثون على العديد من لغات البرمجة الأخرى. لأنه لا يتعين عليهم قضاء الكثير من الوقت في تعلمه. توجد أيضًا حزم في بايثون مصممة خصيصًا لمهام محددة، بما في ذلك pandas و NumPy و SciPy. باختصار، غزت بايثون علم البيانات بسبب الميزات التي ذكرناها أدناه:

- **البساطة:** بايثون هي واحدة من أسهل اللغات للبدء بها. أيضا ، هذه البساطة لا تحد من الميزات التي تحتاجها.
- **المكتبات والأطر:** نظرًا لشعبيتها ، تمتلك بايثون المئات من المكتبات والأطر المختلفة التي تساعد بشكل كبير في عملية التطوير الخاصة بك وتوفر الكثير من الوقت. بصفتك عالم بيانات ، ستجد أن العديد من هذه المكتبات تركز على علم البيانات والتعلم الآلي.
- **مجتمع هائل:** أحد أسباب شهرة بايثون هو أنها تضم مجتمعًا كبيرًا من المهندسين والعلماء. قد تعتقد أن هذا لا ينبغي أن يكون أحد الأسباب الرئيسية لاختيار بايثون ، لكن العكس هو الصحيح. إذا لم تستخدم آراء ودعم الخبراء الآخرين ، فسيكون مسار التعلم الخاص بك صعبًا.
- **اهميتها في التعلم العميق:** تحتوي بايثون على العديد من الحزم مثل keras و Tensorflow و PyTorch التي تساعد علماء البيانات على تطوير خوارزميات التعلم العميق.

- **تمثيل مرئي أفضل للبيانات:** التمثيل المرئي للبيانات هو مفتاح لعلماء البيانات لأنه يساعدهم على فهم البيانات بشكل أفضل. يمكن أن تساعدك بايثون في الرسوم التوضيحية المذهلة مع مكتبات مثل ggplot و Matplotlib و NetworkX وما إلى ذلك.

تثبيت بايتون

التحميل

في هذا القسم، نقدم خطوات تثبيت بايثونفي نظام التشغيل ويندوز. نظرًا لعدم وجود بيئة بايثون مضمنةفي نظام التشغيل ويندوز، يجب تثبيتها بشكل مستقل. يمكن تنزيل حزمة التثبيت من موقع بايثون الرسمي (www.python.org). بعد فتح الموقع الرسمي، ابحث عن شريط التنقل الذي يحتوي على الزر "تنزيل[1]". يوصي موقع الويب برابط افتراضيًا، حيث يمكنه تحديد نظام التشغيل الخاص بك والتوصية بأحدث إصدار من Python 3.x. بعد الدخول إلى صفحة التنزيل الخاصة بالإصدار ذي الصلة، توجد مقدمة أساسية حول البيئة التي تريد تنزيلها. تم تصميم العديد من الإصدارات المختلفة بشكل أساسي لأنظمة التشغيل المختلفة. اعتمادًا على ما إذا كان النظام 32 بت أو 64 بت، يمكنك تحديد ملفات مختلفة لتنزيلها.في الصفحة الجديدة التي تفتح، يمكننا العثور على إصدارات أخرى، بمافي ذلك أحدث إصدار تجريبي والإصدار المطلوب. إذا كنت تريد تثبيت الإصدار 64 بت 3.9.6، فانقر فوق الرابط الموجودفي الصفحة الحالية.

التثبيت

بعد تحميل بايثون، حان وقت تثبيته. يعد تثبيت حزمة ويندوز أمرًا سهلاً للغاية. تمامًا مثل تثبيت برامج ويندوز الأخرى، نحتاج فقط إلى تحديد الخيار المناسب والنقر فوق الزر "التالي" لإكمال التثبيت. عندما تظهر الخيارات أثناء التثبيت، لا تتسرعفي الخطوة التالية. لأنه من أجل الراحةفي المستقبل، عليك اختيار زر "Add Python 3.9.6 to PATH" لمتغير البيئة، يمكن تنفيذ أوامر بايثون مباشرة وبسهولة على سطر أوامر ويندوزفي المستقبل. بعد تحديد " Add Python 3.9.6 to PATH"، حدد التثبيت المطلوب. بالطبع، من الممكن أيضًا تحديد موقع التثبيت، المثبتفي مجلد المستخدم على محرك الأقراص C افتراضيًا. ومع ذلك، من الأفضل معرفة ما هو مجلد المستخدم حتى تتمكن من العثور على ملفات python.exe المثبة عند الحاجة. اتبع التعليمات لتثبيت بايثون بنجاح على نظامك.

[1] download

ابدأ مع بايثون

هناك طريقتان لبدء تشغيل بايثون:

1) **استخدام IDLE الخاص ببايثون.** إذا كنت ترغب في تشغيل بايثون ، يمكنك النقر فوق الزر "ابدأ" على سطح مكتب ويندوز واكتب "IDLE" في مربع "بحث" للدخول بسرعة إلى "read-evaluate-print-loop" بعد تشغيل البرنامج ، سترى صورة مثل الصورة أدناه:

```
Python 3.6.7 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
                                                              Ln: 3  Col: 4
```

IDLE هو IDE (بيئة تطوير متكاملة[1]) لبايثون يوفر محرر واجهة مستخدم رسومي. يبدو تشغيله بسيطًا ومناسبًا للمبتدئين في تعلم لغة بايثون. يوفر IDLE بيئة REPL، أي أنه يقرأ ويقيم ويحسب مدخلات المستخدم ()، ثم يطبع النتيجة ()، وتظهر رسالة "حلقة" (في انتظار الإدخال التالي).

2) **استخدام Windows Prompt.** هناك طريقة أخرى لتشغيل بايثون وهي تشغيل برامج بايثون من سطر أوامر ويندوز. للقيام بذلك ، اضغط على مفتاحي "**Win + R**" لفتح مربع الإشعارات ، ثم أدخل "**cmd**" في المربع الذي يفتح. إذا قمت بتحديد " Add Python 3.x to PATH " عند تثبيت بايثون، فقد تمت إضافة بايثون المثبت إلى متغير بيئة ويندوز. الآن بإدخال كلمة "python" بعد ظهور "**>**"، سيتم تشغيل بايثون بنجاح وسترى صورة مثل الصورة أدناه:

```
Administrator: Command Prompt - python
Microsoft Windows [Version 10.0.18363.628]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>python
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

يشير التعبير "">>>" " إلى أن التثبيت باستخدام بايثون كان ناجحًا وأن بايثون قد بدأت في العمل.

---

[1] Integrated Development Environment

المكتبات وإدارتها في بايتون

المكتباتفي بايثون عبارة عن أجزاء من التعليمات البرمجية يمكن إعادة استخدامها (بمافي ذلك مجموعة من الدوال والطرق) تسمح للمستخدم بأداء العديد من المهام دون أن يضطر المستخدم إلى كتابتها.في بعض الأحيان قد تقضي ساعاتفي كتابة برنامج نصي لمهمة محددة تتطلب وقت تنفيذ $O(n^3)$. ومع ذلك، فمن المحتمل جداً وجود مكتبة لهافي بايثون تحتوي على نفس الوظيفة وتنفذفي $O(n)$.

كمبتدئ ، السؤال الذي يُطرح دائمًا، ما هي المكتبة التي يجب أن أستخدمها للبدء في علم البيانات لجعله ابسط وأسهل في التنفيذ؟ يوجد الآلاف من المكتبات في بايثون. ومع ذلك، في ما يلي ، سنقوم فقط بإدراج أهم مكتبات بايثون في مجال علم البيانات ونوضح استخدام كل منها.



NumPy

NumPy هي مكتبة أساسية للحسابات الرياضية والعلمية. تدعم المكتبة المصفوفات والمصفوفات الكبيرة متعددة الأبعاد، بالإضافة إلى مجموعة كبيرة من الدوال الرياضية عالية المستوى للعمل على هذه المصفوفات. مما لا شك فيه أن NumPy هي مكتبة يجب أن تتعلمها إذا كنت مهتماً بعلم البيانات.

التثبيت:

```
>>> pip install numpy
```



Keras

Keras هي مكتبة بايثون مفتوحة المصدر تُستخدم على نطاق واسع لتعليم نماذج التعلم العميق. توفر هذه المكتبة واجهة لإطار TensorFlow، مما يتيح إجراء تجارب سريعة مع الشبكات العصبية العميقة. بالإضافة إلى ذلك، فإن استخدام هذه المكتبة بسيط للغاية.

التثبيت:

```
>>> pip install keras
```



TensorFlow

TensorFlow هي واحدة من مكتبات بايثون الأكثر استخداماً لمعالجة البيانات والنمذجة ، بالإضافة إلى مكتبة مهمة للتعلم الآلي في بايثون. يعمل TensorFlow على بناءً على الرسوم البيانية لتدفق البيانات التي تحتوي على عُقد وحواف. نظراً لأن آلية التنفيذ رسومية، فمن الأسهل بكثير تنفيذ كود TensorFlow أثناء استخدام وحدة معالجة الرسومات.

التثبيت:

```
>>> pip install tensorflow
```



PyTorch

PyTorch هو إطار عمل تعلم آلي وتعلم عميق مفتوح المصدر طوره باحثو Facebook AI. يستخدم العديد من علماء البيانات حول العالم PyTorch على نطاق واسع لمعالجة مسائل اللغة الطبيعية ورؤية الكمبيوتر. يمكن لعلماء البيانات تصميم الرسوم البيانية الحسابية ديناميكيًا عبر PyTorch.

التثبيت:

```
>>> pip install torch torchvision torchaudio
```



Scrapy

تعد Scrapy واحدة من أكثر مكتبات بايثون شيوعًا لاستخراج البيانات من مواقع الويب. تساعد هذه المكتبة فى استرداد البيانات من مواقع الويب بطريقة فعالة. يساعد Scrapy فى استرداد البيانات المنظمة من الويب والتى يمكن استخدامها لاحقًا لنمذجة التعلم الآلي الخاص بک.

التثبيت:

```
>>> pip install Scrapy
```



BeautifulSoup

BeautifulSoup هى واحدة من أفضل مكتبات برامج زاحف الويب Web crawler وأكثرها شيوعًا والتى يمكن استخدامها لاستخراج البيانات من ملفات HTML و XML. تساعدک هذه الأداة فى تنظيف وتحليل المستندات المستخرجة من الويب. هذا يوفر ساعات أو أيام للمبرمجين والمحللين.

التثبيت:

```
>>> pip install beautifulsoup4
```

Pandas

تعد Pandas واحدة من أهم مكتبات علم البيانات المستخدمة لإنشاء هياكل البيانات. يوفر Pandas مرونة قوية فى إنشاء هياكل البيانات لعلم البيانات. لأنه يمكن أن يخلق هياكل بيانات متعددة الأبعاد، مهيكلة، غير متجانسة، إلخ. بالإضافة إلى ذلک، تُستخدم هذه المكتبة لتحليل القواعد النحوية والبيانات.

التثبيت:

```
>>> pip install pandas
```



SciPy

SciPy هى مكتبة بايثون أخرى تُستخدم لحل المسائل العلمية والرياضية وهى مبنية على المكون الإضافى NumPy. تعد الحوسبة الرقمية جانبًا مهمًا من علم البيانات، ويمكن لـ SciPy توجيه علماء البيانات فى حل المسائل الرياضية المعقدة. يمكن القول إن SciPy هو إصدار متقدم من NumPy يحتوى على ميزات إضافية مثل النسخة الجبرية الكاملة. SciPy سريع ولديه قدرة حوسبة عالية.

التثبيت:

```
>>> pip install scipy
```



Scikit-Learn

يستخدم علماء البيانات Scikit-Learn لنمذجة البيانات إحصائياً بما فى ذلك التصنيف وتقليل الأبعاد والتجميع والتوقع. يعتمد Scikit-Learn على مكتبات NumPy وMatplotlib. يعد تقليل حجم البيانات أحد أكثر الميزات المفيدة فى Scikit-Learn. لأن البيانات الناتجة ستكون أقل تعقيداً.

التثبيت:

```
>>> pip install scikit-learn
```



Matplotlib

يعد رسم المخططات أحد الخطوات الأساسية أثناء تحليل البيانات وإدارتها. تعد Matplotlib واحدة من أكثر المكتبات شيوعاً فى مجتمع بايثون للرسم والتمثيل المرئى للبيانات الثابتة والمتحركة والتفاعلية.

التثبيت:

```
>>> pip install matplotlib
```



Seaborn

Seaborn هى مكتبة بايثون تعتمد على Matplotlib وتستخدم على نطاق واسع لرسم البيانات. يمكن لعلماء البيانات استخدام هذه المكتبة لإنشاء خرائط حرارية[1]. عدد الخيارات التى توفرها Seaborn لتصور البيانات هائل.

التثبيت:

```
>>> pip install seaborn
```



NetworkX

NetworkX هى مكتبة بايثون لإنشاء ومعالجة وهيكلة وتحليل الشبكات المعقدة الكبيرة. توفر NetworkX ، بالإضافة إلى الرسوم التوضيحية الممتازة ثنائية وثلاثية الأبعاد ، للمستخدم العديد من المقاييس والخوارزميات القياسية للرسم البيانى ، مثل أقصر مسار ومركز وترتيب الصفحة والمزيد.

التثبيت:

```
>>> pip install networkx
```



NLTK

تعد مجموعة أدوات اللغة الطبيعية، والمختصرة NLTK ، مكتبة بايثون مهمة يستخدمها علماء البيانات لمجموعة متنوعة من مهام معالجة اللغة الطبيعية ، مثل وضع علامات على النص ، والترميز ، والتجذير ، والمزيد.

التثبيت:

```
>>> pip install nltk
```

[1] Heat Map

تثبيت وحذف وتحديث المكتبات

يجب عليك استخدام Pip لإدارة مكتبات Pip. Pip هي أداة أساسية تتيح لك تنزيل الحزم التي تحتاجها وتحديثها وحذفها. بالإضافة إلى ذلك، يمكن استخدامه للتحقق من التبعيات المناسبة والتوافق بين الإصدارات.

يتم تثبيت مكتبة باستخدام Pipفي سطر أوامر ويندوز. على سبيل المثال، افترض أننا نريد تثبيت مكتبة NumPy. توضح الخطوات التالية كيفية تثبيت هذه المكتبة:

- اضغط أولاً على مفتاحي "Win + R" لفتح مربع الإشعارات ثم أدخل "cmd" في المربع الذي يفتح. ثم أدخل الأمر التالي في سطر الأوامر:

```
> pip install numpy
```

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.18363.628]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>pip install numpy
```

- للتأكد من تثبيت المكتبة ، قم بتشغيل سطر أوامر بايثون واكتب الأمر التالي:

```
>>> import numpy
```

- لن يتم عرض أي رسالة إذا تم تثبيت المكتبة بشكل صحيح. إذا لم يتم تثبيت المكتبة على جهاز الكمبيوتر الخاص بك ، فسترى هذه الرسالة عن طريق تنفيذ الأمر أعلاه:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named numpy
```

يتم استخدام الأمر التالي لحذف مكتبة (على سبيل المثال، numpy):

```
> pip uninstall numpy
```

في بعض الأحيان، تجد نفسك في موقف يتعين عليك فيه ترقية مكتبة. نظرًا لأن تثبيت مكتبة أخرى يتطلب إصدارًا أحدث من المكتبة مثبتًا على جهاز الكمبيوتر الخاص بك، فقد ترغب في

الاستفادة من الإصدار المحدث، الذي يحتوي على ميزات إضافية. لذلك، لتحديث مكتبة (على سبيل المثال مكتبة numpy)، قم بتشغيل الأمر التاليفي سطر الأوامر:

```
> pip install --upgrade numpy
```

يمكن القيام بذلك لعرض النسخة المثبتة من المكتبة:

```
>>> import numpy
>>> numpy.__version__
'1.19.2'
```

نوتبوك جوبيتر Jupyter Notebook

نوتبوك جوبيتر Jupyter Notebook هو أداة قوية بشكل لا يصدق لتطوير وتقديم مشاريع علم البيانات التفاعلية التي يمكن أن تتضمن نصًا أو صورة أو صوتًا أو فيديو بالإضافة إلى تنفيذ التعليمات البرمجية. يجمع النوتبوك Notebook بين التعليمات البرمجية والمخرجات مع الرسوم التوضيحية والنص السردي والمعادلات الرياضية والوسائط الأخرىفي مستند واحد. بمعنى آخر، النوتبوك هو مستند واحد يمكنك من خلال نوتبوك جوبيتر تنفيذ التعليمات البرمجية وعرض الإخراج وإضافة الأوصاف والصيغ والرسوم التخطيطية لجعل عملك أكثر وضوحًا وقابلية للفهم وقابلية للتكرار والمشاركة. منصات أخرى مماثلة، مثل Spyder، توفر للمستخدمين نوافذ متعددة تعقد الأمور. أصبحت نوتبوك جوبيتر واجهة مستخدم قوية وفعالة من خلال توفير نافذة واحدة للمستخدم يتم فيها عرض الأجزاء المنفذة من التعليمات البرمجية وإخراجها داخليًا. يتيح ذلك للمستخدمين تطوير التعليمات البرمجية بكفاءة، بالإضافة إلى النظر إلى العمل السابق كمرجع وحتى إجراء تغييرات عليه.في هذا القسم، سوف نعلمك كيفية تثبيت واستخدام نوتبوك جوبيتر لمشاريع علم البيانات.

تثبيت Jupyter

لتثبيت نوتبوك جوبيتر، تحتاج إلى تثبيت بايثون مسبقًا. حتى إذا كنت تخطط لاستخدام نوتبوك جوبيتر للغات البرمجة الأخرى، فإن بايثون هي الركيزة الأساسية لنوتبوك جوبيتر. لتثبيت جوبيتر، ما عليك سوى كتابة الأمر التاليفي سطر أوامر ويندوز:

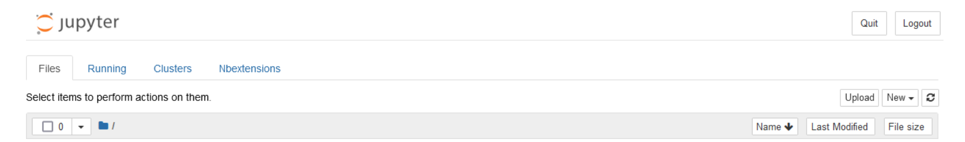
```
> pip install jupyter
```

تشغيل نوتبوك جوبيتر وإنشاء نوتبوك جديد

لتشغيل نوتبوك جوبيتر، افتح سطر الأوامر واكتب الأمر التالي فيه:

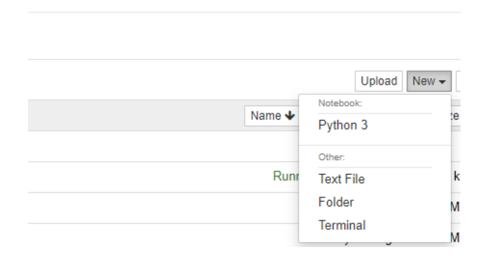
```
> jupyter notebook
```

بعد تنفيذ الأمر أعلاه، سيتم تشغيل متصفح الويب الافتراضي الخاص بك مع جوبيتر. عند تشغيل نوتبوك جوبيتر، انتبه إلى دليل سطر الأوامر، حيث يصبح هذا الدليل هو الدليل الرئيسي الذي يظهر على الفور على نوتبوك جوبيتر وسيكون لديك حق الوصول إلى الملفات والأدلة الفرعية الموجودة فيه فقط. قم بتشغيل أمر نوتبوك جوبيتر وسترى صفحة مثل الصفحة أدناه:



ومع ذلك، فإن هذه الصفحة ليست نوتبوك بعد وهي سطح المكتب الوحيد من جوبيتر المصمم لإدارة نوتبوك جوبيتر الخاصة بك واستخدامها كدليل لاستكشاف[1] وتحرير وإنشاء النوتبوك الخاصة بك. تعتمد النوتبوك وسطح المكتب من جوبيتر على متصفح الويب، ويقوم جوبيتر بإعداد خادم بايثون محلي لتوصيل هذه التطبيقات بمتصفح الويب الخاص بك.

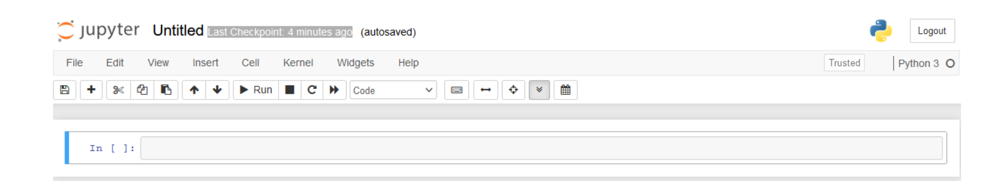
لإنشاء نوتبوك جديد، انتقل إلى الدليل حيث تريد إنشاء النوتبوك الأول لك وانقر على زر القائمة المنسدلة "جديد" أعلى يمين سطح المكتب وحدد "Python 3":



بعد ذلك، سيتم فتح النوتبوك الأول لكفي علامة تبويب جديدة[2] مثل تلك الموجودة أدناه:

[1] exploring

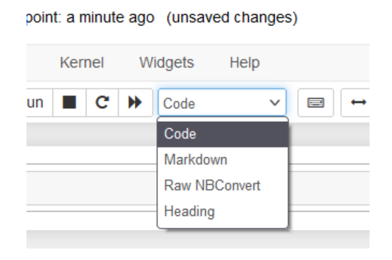
[2] new tab



إذا عدت إلى سطح مكتب جوبيتر، فسترى ملف Untitled.ipynb الجديد وسترى نصًا أخضر يخبرك أن النوتبوك الخاص بك قيد التشغيل.

العمل على نوتبوك جوبيتر

النوتبوك مكون من خلايا. المربعات التي تحتوي على كود أو نص يمكن قراءته من قبل الإنسان. تحتوي كل خلية من هذه الخلايا على نوع[1] يمكن تحديده من خيارات القائمة المنسدلة:



الخيار الافتراضي للخلايا هو "code". لإنشاء خلية يمكن قراءتها بواسطة الإنسان (نص)، يجب استخدام نوع الخلية "Markdown" وكتابته وفقًا لقواعد تنسيق Markdown.

دعنا نختبر كيفية أداء الخلية بمثال كلاسيكي. اكتب ("Hello World!") print في خلية وانقر على الزر ▶ Run في شريط الأدوات العلوي أو اضغط على Ctrl + Enter. ستكون النتيجة على هذا النحو:

```
In [1]: print('Hello World!')
        Hello World!
```

[1] type

Markdown هي لغة ترميز خفيفة الوزن لتنسيق النص البسيط. تركيبته[1] متوافقة مع علامات[2] HTML. نحن نغطي الأساسيات بمثال سريع ، من خلال وضع نص مثل ذلك الموجود في الصفحة التالية في خلية تم تحديد نوع Markdown بالفعل ، سترى النتيجة في الصورة التالية بعد التنفيذ. ومع ذلك ، تجدر الإشارة إلى أنه بمجرد التنفيذ ، يحول هذا الإخراج المعروض نفس الخلية إلى نص.

```
# This is a level 1 heading
## This is a level 2 heading
This is some plain text that forms a paragraph. Add emphasis via **bold** and __bold__, or *italic* and _italic_.

Paragraphs must be separated by an empty line.

* Sometimes we want to include lists.
* Which can be bulleted using asterisks.

1. Lists can also be numbered.
2. If we want an ordered list.

[It is possible to include hyperlinks](https://www.example.com)
```

This is a level 1 heading

This is a level 2 heading

This is some plain text that forms a paragraph. Add emphasis via **bold** and **bold**, or *italic* and *italic*.

Paragraphs must be separated by an empty line.

- Sometimes we want to include lists.
- Which can be bulleted using asterisks.

1. Lists can also be numbered.
2. If we want an ordered list.

[It is possible to include hyperlinks](https://www.example.com)

انقر فوق الزر + لإنشاء خلية جديدة على جوبيتر، عن طريق شريط القوائم للصفحة او اضغط على b + esc عبر اختصار لوحة المفاتيح. يؤدي هذا إلى إنشاء خلية جديدة (افتراضيًا من نوع Code) ضمن الخلية المحددة حاليًا.

---

[1] syntax

[2] tags

البرمجة بلغة بايثون

تركيب الجملة[1] في بايثون

تم تطوير لغة بايثونفي الأصل كلغة تعليمية، ولكن سهولة استخدامها وتركيب الجملة النظيف جعلها شائعة لدى المبتدئين والمحترفين على حدٍ سواء. دفع بناء جملة بايثون النظيف بالبعض إلى تسميته "الشفرة الزائفة التنفيذية[2]". غالبًا ما تكون قراءة وفهم سطر من بايثون أسهل بكثير من قراءة سطر مشابهفي C، على سبيل المثال.

يشير تركيب الجملة في لغة البرمجة إلى بنية اللغة. بمعنى آخر، التركيب اللغوي هو مجموعة من القواعد التي تحدد كيفية كتابة البرمجة بلغة ما.

الكلمات المحجوزة

الكلمات المفتاحية هي بعض الكلمات المحجوزةفي بايثون والتي لها معاني خاصة. تستخدم الكلمات المفتاحية لتعريف القواعد النحوية وهيكل اللغة. لا يمكن استخدام الكلمة المفتاحية كمعرف أو وظيفة أو اسم متغير. الكلمات المفتاحيةفي بايثون حساسة لحالة الأحرف، لذا يجب كتابتها كما هي. جميع الكلمات المفتاحيةفي بايثون، باستثناء True وFalse وNone، هي أحرف صغيرة. قد تتغير الكلمات المفتاحيةفي إصدارات مختلفة من بايثون. قد يتم إضافة بعض الإضافات أو قد تتم إزالة البعض الآخر. يمكنك دائمًا الحصول على قائمة بالكلمات المفتاحية في إصدارك الحالي عن طريق كتابة الأوامر التالية:

```
In  [1]:  import keyword
          keyword.kwlist
Out [1]:  ['False','None','True','and', 'as','assert','async','await',
          'break','class','continue','def','del','elif','else','except',
           'finally','for','from','global','if','import', 'in','is', 'lambda',
           'nonlocal', 'not','or', 'pass', 'raise', 'return', 'try', 'while',
           'with', 'yield']
```

المعرف

المعرف هو الاسم الذي نستخدمه لتحديد متغير أو دالة أو فئة أو وحدة نمطية أو كائن. هذا يساعد على تمييز كيان واحد عن الآخر.

---

[1] Syntax

[2] executable pseudocode

قواعد كتابة المعرف

هناك بعض القواعد لكتابة المعرفات. بادئ ذي بدء، نحتاج إلى معرفة أن لغة بايثون حساسة لحالة الأحرف. هذا يعني أن Name وname هما معرفان مختلفانفي بايثون. المدرجة أدناه هي بعض القواعد لكتابة المعرفاتفي بايثون:

1. **يمكن أن تكون المعرفات مزيجًا من الأحرف الصغيرة (من a الى z) أو الأحرف الكبيرة (من A الى Z) أو الأرقام (من 0 الى 9) أو الشرطة السفلية [1] (_). أحرف مثل myPython، my_Python و var_1 كلها صالحة.**
2. **لا يمكن أن يبدأ المعرف برقم.**
3. **لا يمكن استخدام رموز معينة مثل!، @، #، $، % الخ كمعرفات.**
4. **يمكن أن يكون المعرف بأي طول.**

المتغير

المتغير هو موقع يستخدم لتخزين البياناتفي الذاكرة. هذا يعني أنه عند إنشاء متغير، فإنك تشغل بعض المساحةفي الذاكرة. يتم تعيين اسم لكل متغير بحيث يمكن تحديده والوصول إليه من متغيرات أخرى.

اسناد قيم للمتغيرات

يعد إنشاء المتغيراتفي بايثون أمرًا بسيطًا، ما عليك سوى كتابة اسم المتغير على اليسار = وقيمة المتغير على اليمين:

```
In [1]: num = 5
In [2]: str = "Python"
```

لا تحتاج إلى تحديد نوع المتغير، تستنتج بايثون النوع بناءً على القيمة التي نخصصها.

الاسناد المتعدد

تسمح لنا بايثون بتعيين قيمة لمتغيرات متعددةفي الذاكرة، يسمى الاسناد المتعدد. يمكننا تطبيق اسناد متعدد بطريقتين، إما عن طريق تعيين قيمة واحدة لمتغيرات متعددة أو عن طريق تعيين قيم متعددة لمتغيرات متعددة. تأمل الأمثلة التالية:

```
In [3]:   a=b=c=20
          print("a:",a)
          print("b:",b)
          print("c:",c)
Out [3]:  a: 20
```

[1] underscore

```
        b: 20
        c: 20
In [4]: a,b,c=1,2.54,"python"
        print("a:",a)
        print("b:",b)
        print("c:",c)
Out [4]: a: 1
        b: 2.54
        c: python
```

أنواع البيانات

في البرمجة، يعد نوع البيانات مفهومًا مهمًا. يمكن للمتغيرات تخزين البيانات من أنواع مختلفة، ويمكن للأنواع المختلفة القيام بأشياء مختلفة. نظرًا لأن كل شيءفي برمجة بايثون هو كائن، فإن أنواع البيانات هيفي الواقع فئات[1]، والمتغيرات هي أمثلة للكائن.في هذا القسم، يتم فحص نوع البيانات **الرقمية** ونوع البيانات **النصية** فقطفي بايثون، ويتم وصف الأنواع الأخرىفي قسم هياكل البيانات.

الاعداد

في بايثون، يشير نوع البيانات الرقمية إلى البيانات التي تحتوي على قيمة رقمية. يمكن أن تكون القيم الرقمية أعدادًا صحيحة أو أرقامًا فاصلة عائمة أو حتى أرقامًا معقدة. يتم تعريف هذه القيم على أنها فئات int وfloat و complex في بايثون. يتم استخدام وظيفة type() لتحديد نوع البيانات.

```
In [1]: a = 6
        b = 7.0
        c = 4 + 3j
        print("Type of a: ", type(a))
        print("Type of b: ", type(b))
        print("Type of c: ", type(c))
Out [1]: Type of a:  <class 'int'>
        Type of b:  <class 'float'>
        Type of c:  <class 'complex'>
```

[1] Classes

السلاسل النصية

في بايثون، السلاسل عبارة عن مصفوفات من البايتات التي تمثل أحرف Unicode. السلسلة النصية عبارة عن مجموعة من حرف واحد أو أكثر. لا توجد بيانات حرفيةفي بايثون، فالحرف عبارة عن سلسلة بطول واحد.

```
In  [2]:  Str_1 = 'Python Data Types'
          Str_1
Out [2]:  'Python Data Types'
In  [3]:  Str_2 = "data science"
          Str_2
Out [3]:  "data science"
In  [3]:  type(Str_2)
Out [3]:  str
In  [3]:  type("c")
Out [3]:  str
```

العوامل

تُستخدم عوامل[1] بايثون عمومًا لإجراء عمليات على القيم والمتغيرات. إنها رموز قياسية تُستخدم لأداء العمليات المنطقية والحسابية والمقارنة.

العوامل الحسابية

تُستخدم العوامل الحسابية لإجراء عمليات حسابية مثل الجمع والطرح والضرب والقسمة.

العملية	شرح العامل	العامل
x + y	الجمع	+
x – y	الطرح	–
x * y	الضرب	*
x / y	القسمة (قسمة x على y)	/
x // y	القسمة (حاصل القسمة التحتية x على y)	//
x % y	باقي القسمة	%
x ** y	القوة	**

```
In  [1]:  x=10
          y=3
          add = x+y
          sub = x - y
          mul = x * y
          div1 = x / y
          div2 = x // y
```

[1] operators

```
        mod = x % y
        p = x ** y
        print("Addition:",add)
        print("Subtraction:",sub)
        print("Multiplication:",mul)
        print("Division(float):",div1)
        print("Division(floor):",div2)
        print("Modulo:",mod)
        print("Power:",p)
Out [1]: Addition: 13
        Subtraction: 7
        Multiplication: 30
        Division(float): 3.3333333333333335
        Division(floor): 3
        Modulo: 1
        Power: 1000
```

العوامل المنطقية

يتم تطبيق العوامل المنطقية (and، or و not) على التعبيرات المنطقية وتكون النتيجة صواب أو خطأ. يتم استخدام هذه العوامل للشروط المعقدة.

العامل	شرح العامل	العملية
and	True إذا كان كلا المعاملين True	x and y
or	True إذا كان أحد المعاملات True	x or y
not	إذا كان True فالمعامل False	not x

```
In  [1]: X = True
         Y = False
         print(X and Y)
         print(X or Y)
         print(not X)
Out [1]: False
         True
         False
```

عوامل المقارنة

تقارن هذان العاملان قيم جانبيهما ويحددان العلاقة بينهما. تسمى عوامل المقارنة أيضًا بالعوامل الارتباطية.

العامل	شرح العامل	العملية
>	أكبر من: True عندما يكون المعامل الأيسر أكبر من المعامل الأيمن.	x > y
<	أصغر من: True عندما يكون المعامل الأيسر أصغر من المعامل الأيمن.	x < y

x == y	يساوي: True إذا كان كلا المعاملين متساويين.	==
x! = y	غير مساوي: True إذا كان كلا المعاملين غير متساويين.	!=
x >= y	أكبر من او يساوي: إذا كان المعامل الأيسر أكبر من أو يساوي المعامل الأيمن.	>=
x <= y	أصغر من يساوي: إذا كان المعامل الأيسر أقل من أو يساوي المعامل الأيمن.	<=

```
In  [1]:  a = 11
          b = 26
          print(a > b)
          print(a < b)
          print(a == b)
          print(a != b)
          print(a >= b)
          print(a <= b)
Out [1]:  False
          True
          False
          True
          False
          True
```

هياكل البيانات

يعد تنظيم البيانات وإدارتها وتخزينها أمرًا مهمًا لأنه يتيح الوصول السهل والتغيير الفعال. تسمح لك هياكل البيانات بتنظيم بياناتك بطريقة يمكنك من خلالها تخزين مجموعة من البيانات وربطها معًا وتنفيذ العمليات بناءً عليها. هياكل البيانات هي الكتل الأساسية لحل المشكلات بكفاءةفي العالم الحقيقي. إنها أدوات مُثبتة ومحسّنة تمنحك إطارًا سهلاً لتنظيم التطبيقات.

تدعم بايثون ضمنيًا هياكل البيانات التي تسمح لك بتخزين البيانات والوصول إليها. يوجد إجمالي أربعة هياكل بيانات داخليةفي لغة برمجة بايثون. تتضمن هياكل البيانات هذه **القوائم والصفوف والقواميس والمجموعات**. هياكل بيانات بايثون بسيطة ولكنها قوية جدًا. يعد إتقان استخدامها جزءًا مهمًا من أن تصبح مبرمجًا ماهرًافي لغة بايثون. بالإضافة إلى ذلك، يبحث أصحاب العمل والموظفون عن وقت التنفيذ وكفاءة الموارد عند حل مشكلات البرمجةفي العالم الحقيقي. إن معرفة هياكل البيانات الأنسب للحل الحالي يزيد من أداء البرنامج ويقلل من الوقت اللازم لإنشائه. لهذا السبب، تحتاج معظم الشركات الكبرى إلى فهم قوي لهياكل البيانات واختبارها بدقةفي مقابلات البرمجة الخاصة بهم.

تساعد هياكل البيانات فيما يلي:

- إدارة واستخدام مجموعة البيانات.
- البحث السريع عن بيانات محددة فى قاعدة البيانات.
- إنشاء اتصالات هرمية أو علاقات بين نقاط البيانات.
- تبسيط وتسريع معالجة البيانات.

القوائم

يتم تعريف القائمة (list) على أنها مجموعة مرتبة من العناصر. بمعنى آخر، يحتوي على قائمة متسلسلة من العناصر. ترتيب العناصر هو خاصية متأصلة تظل ثابتة طوال عمر القائمة. نظرًا لأن كل شيءفي بايثون عبارة عن كائن، فإن إنشاء قائمة يؤدي بشكل أساسي إلى إنشاء كائن بايثون من نوع معين. عند إنشاء قائمة، يجب وضع جميع العناصرفي القائمةفي [] ومفصولة بفواصل لإخطار بايثون بأن القائمة قد تم إنشاؤها. يمكن إنشاء قائمةفي بايثون على النحو التالي:

```
List_A = [item 1, item 2, item 3, ...., item n]
```

```
In  [1]: my_list = [1, 2, 3, 4]
In  [2]: my_list
Out [2]: [1, 2, 3, 4]
```

إذا لم تضع أي عناصر داخل القوس، فستتلقى قائمة فارغة كإخراج:

```
In  [1]: my_list1 = []
In  [2]: my_list1
Out [2]: []
```

يمكن أن تكون كل قائمة عددًا من العناصر بأنواع مختلفة من البيانات:

```
In  [3]: my_list = [1, 'example', 5.45]
In  [4]: my_list
Out [4]: [1, 'example', 5.45]
```

بالإضافة إلى ذلك، يمكن أن تحتوي القائمة على قائمة أخرى كعنصر. تُعرف هذه القائمة بالقائمة المتداخلة:

```
In  [5]: my_list1 = [1.56, 'python']
In  [6]: my_list2 = ['example', 1]
In  [7]: my_list = [5, my_list1 ,'data scientist', my_list2]
In  [8]: my_list
Out [6]: [5, [1.56, 'python'], 'data scientist', ['example', 1]]
```

اضافة عنصر

يمكن إضافة عنصر إلى قائمةفي بايثون باستخدام الدوال append() و extend() و insert():

- الدالة **append()** يضيف كل العناصر المنقولة إلى القائمة كعنصر واحد.
- الدالة **extend()** يضيف عناصر إلى القائمة واحدة تلو الأخرى.
- الدالة **insert()** يضيف عنصرًا في فهرس معين إلى القائمة.

```
In  [7]: my_list = [7, 2, 1]
In  [8]: my_list
Out [8]: [7, 2, 1]
In  [9]: my_list.append([44, 15,'python'])
In  [10]: my_list
Out [10]: [7, 2, 1, [44, 15, 'python']]
In  [11]: my_list.extend(['example',2])
In  [12]: my_list
Out [12]: [7, 2, 1, [44, 15, 'python'], 'example', 2]
In  [13]: my_list.insert(1, 'insert_example1')
In  [14]: my_list.insert(6, 'insert_example2')
In  [15]: my_list
Out [15]: [7, 'insert_e1', 2, 1, [44, 15, 'python'], 'example',
'insert_e2', 2]
```

حذف عنصر

يمكن إزالة عنصر من قائمة في بايثون باستخدام الدوال clear()، pop()، remove() وdel():

- باستخدام الدالة **clear()** ، يتم حذف جميع عناصر القائمة.
- تحذف الدالة **pop()** عنصرًا بناءً على الفهرس وتعرض قيمته في المخرجات.
- باستخدام الدالة **remove()** يمكن إزالة عنصر بناءً على قيمته.
- باستخدام الدالة **del()** يمكن حذف عناصر المصفوفة بناءً على الفهرس. الفهرس الأول هو 0 والفهرس الأخير هو 1-.

```
In  [16]: my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
In  [17]: my_list
Out [17]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
In  [18]: my_list.clear()
Out [18]: []
In  [19]: my_list = [8, 2, 3, 7, 9, 1]
```

```
In  [20]: my_list.pop(0)
Out [20]: 8
In  [21]: my_list.pop(4)
Out [21]: 1
In  [22]: my_list = [12, 1, 5, 2, 4]
In  [23]: my_list.pop(-3)
Out [23]: 5
In  [24]: my_list = ['Ali', 'Mohammad', 'Milad', 1, 5.69]
In  [25]: my_list.remove('Milad')
In  [25]: my_list
Out [25]: ['Ali', 'Mohammad', 1, 5.69]
In  [26]: my_list.remove(1)
In  [27]: my_list
Out [27]: ['Ali', 'Mohammad', 5.69]
In  [28]: my_list = [4, 7, 5, 1, 4]
In  [29]: del my_list[0]
In  [30]: my_list
Out [30]: [7, 5, 1, 4]
In  [31]: del my_list[-1]
In  [32]: my_list
Out [32]: [7, 5, 1]
```

دوال اخرى

هناك العديد من الدوال الأخرى التي يمكن استخدامها عند العمل مع القوائم:

- **ترجع الدالة len() طول القائمة.**
- **ترجع الدالة index() فهرس أحد العناصر (ملاحظة: إذا ظهر عنصر في القائمة عدة مرات ، فسيتم إرجاع الفهرس الأول المطابق).**
- **باستخدام الدالة sort() يتم فرز القائمة بترتيب صعودي.**

```
In  [33]: my_list1 = [4, 7, 5, 1, 4, 12]
In  [34]: len(my_list)
Out [34]: 6
In  [35]: my_list.index(5)
Out [35]: 2
```

```
In  [36]: my_list.sort()
In  [37]: my_list
Out [37]: [1, 4, 4, 5, 7, 12]
```

الصفوف

الصفوف (Tuples) هي سلسلة ثابتة وغير قابلة للتغيير من العناصرفي بايثون. أهم فرق بينها وبين القوائم هو ثباتها.في حين أن القوائم قابلة للتغيير، لا تتوفر هذه الميزةفي الصفوف.

أسهل طريقة لإنشاء صف هي كما يلي:

```
tuple_A = item 1, item 2, item 3,…, item n
```

يعد استخدام الأقواس في تكوين المجموعة أمرًا اختياريًا ، لكن يوصى بالتمييز بين بداية المجموعة ونهايتها:

```
tuple_A = (item 1, item 2, item 3,…, item n)
```

```
In  [1]: my_tuple = (1, 2, 3)
In  [2]: my_tuple
Out [2]: (1, 2, 3)
```

باستدعاء tuple، يمكنك تحويل كل سلسلة إلى عدة صفوف:

```
In  [3]: tuple([1, 3, 8])
Out [3]: (1, 3, 8)
In  [4]: tup_A = tuple('Python')
In  [5]: tup_A
Out [5]: ('P', 'y', 't', 'h', 'o', 'n')
```

الدوال

هناك العديد من الدوال التي يمكن استخدامها عند العمل مع الصفوف:

- **ترجع ()len الدالة طول المجموعة.**
- **الدالة ()index ترجع فهرس العنصر.**
- **تُرجع الدالة ()max أكبر قيمة في المجموعة.**
- **تُرجع الدالة ()min أصغر قيمة في المجموعة.**

```
In  [6]: tup_A = tuple('Python')
In  [7]: tup_A.index('y')
Out [7]: 1
In  [8]: my_tuple = (8, 1, 4, 5, 3)
In  [9]: max (my_tuple)
Out [9]: 8
In  [9]: min (my_tuple)
Out [9]: 1
```

عند كتابة صف بعنصر واحد، يجب استخدام فاصلة بعد العنصر. يمكن إنشاء صف تحتوي على عنصر واحد على النحو التالي:

```
tuple_A = (item 1,)
```

لإنشاء صف فارغ، يجب على المستخدم إنشاء زوج من الأقواس الفارغة على النحو التالي:

```
tuple_A = ()
```

```
In  [10]: Empty_tuple= ()
In  [11]: Empty_tuple
Out [11]: ()
```

لماذا يفضل الصف Tuple على القائمة List؟

- تعتبر الصفوف أسرع من القوائم. يتم تخزين المجموعة في كتلة واحدة من الذاكرة. تعتبر المجموعات غير قابلة للتغيير ، لذلك ليست هناك حاجة إلى مساحة إضافية لتخزين العناصر الجديدة.
- يُفضل الصف عندما لا يرغب المستخدم في تغيير البيانات. في بعض الأحيان ، يريد المستخدم إنشاء كائن يظل سليماً طوال حياته. تعتبر المجموعات غير قابلة للتغيير، لذا يمكن استخدامها لمنع الإضافة أو التعديل أو الحذف العرض للبيانات.

القواميس

القاموس (Dictionary)في بايثون هو مجموعة غير مرتبة من القيم والتي على عكس هياكل البيانات الأخرى التي تحتوي على قيمة واحدة فقط كعنصر، يتم استخدامها لتخزين أزواج القيمة

والمفتاح. يتم توفير قيمة المفتاحفي القاموس لمزيد من التحسين.في القاموس، يجب أن تكون المفاتيح فريدة.

```
my_dictionary= {key 1 : value 1,  key 2 : value 2}
```

```
In  [1]: my_dict = {'First': 'Python', 'Second': 'Julia'}
In  [2]: my_dict
Out [2]: {'First': 'Python', 'Second': 'Julia'}
```

لتغيير قيم القاموس، عليك القيام بذلك باستخدام المفاتيح. لذلك، قم أولاً بالوصول إلى المفتاح ثم قم بتغيير قيمته:

```
In  [3]: my_dict['Second'] = 'golang'
In  [4]: my_dict
Out [4]: {'First': 'Python', 'Second': 'golang'}
```

لإضافة قيم، ما عليك سوى إضافة زوج قيم_ مفتاح آخر وفقًا للأمر التالي:

```
In  [3]: my_dict['Third'] = 'Rust'
In  [4]: my_dict
Out [4]: {'First': 'Python', 'Second': 'golang', 'Third': 'Rust'}
```

تُستخدم الدلة pop() لحذف قيمة (تُرجع هذه الدالة أيضًا القيمة المحذوفة):

```
In  [5]: my_dict.pop('Third')
Out [5]: Rust
In  [6]: my_dict
Out [6]: {'First': 'Python', 'Second': 'golang'}
```

يتم استخدام الدلة clear() لمسح القاموس بأكمله:

```
In  [7]: my_dict.clear()
In  [8]: my_dict
Out [8]: {}
```

للوصول إلى عنصر، ما عليك سوى الضغط على مفتاحه:

```
In  [9]: my_dict = {'First': 'Python', 'Second': 'Julia'}
```

```
In  [10]: my_dict['First']
Out [10]: Python
```

يتم استخدام دالة keys() لعرض المفاتيح ويتم استخدام دالة values() لعرض القيم. يمكن عرض زوج المفتاح والقيمة باستخدام دالة items():

```
In  [11]: my_dict = {'First': 'Python', 'Second': 'Julia'}
In  [12]: my_dict.keys()
Out [12]: dict_keys(['First', 'Second'])
In  [13]: my_dict.values()
Out [13]: dict_values(['Python', 'Julia'])
In  [14]: my_dict.items()
Out [14]: dict_items([('First', 'Python'), ('Second', 'Julia')])
```

المجموعات

يتم تعريف المجموعة (Set) على أنها مجموعة من العناصر الفريدة التي لا تتبع ترتيبًا معينًا. تُستخدم المجموعات عندما يكون وجود كائنفي مجموعة من الكائنات أكثر أهمية من عدد المرات التي تظهر فيها الكائنات أو ترتبها.في المجموعات، إذا تكررت البيانات أكثر من مرة، يتم إدخالهافي المجموعة مرة واحدة فقط. على عكس الصفوف، فإن المجموعات قابلة للتغيير؛ أي أنه يمكن تعديلها أو إضافتها أو استبدالها أو إزالتها. يمكن عرض مجموعة مثال على النحو التالي:

```
set_a = {"item 1", "item 2", "item 3",....., "item n"}
```

```
In  [1]: my_set = {2, 2, 3, 1, 4, 5, 5, 5}
In  [2]: my_set
Out [2]: {1, 2, 3, 4, 5}
```

يمكنك استخدام الدالة add() لإضافة عنصر:

```
In  [3]: my_set = {8, 1, 5}
In  [4]: my_set.add(6)
In  [5]: my_set
Out [5]: {1, 5, 6, 8}
```

هنالك عمليات تطبق على مجموعات الرياضيات مثل الاتحاد (Union) والاشتراك (Interaction) وما إلى ذلك. يوضح المثال التالي المجموعة المكونة من اتحاد مجموعتين:

```
In  [3]: a = {1, 2, 3, 4, 5}
In  [4]: b = {6, 4, 5, 1, 3, 8, 7}
In  [5]: a.union(b)
Out [5]: {1, 2, 3, 4, 5, 6, 7, 8}
```

هياكل التحكم والحلقات

في الحياة اليومية ، نتخذ القرارات كل يوم ونتخذ الخطوات التالية بناءً على القرارات التي نتخذها. لذا فإن جميع أنشطتنا اليومية تعتمد على القرارات التي نتخذها. ينشأ موقف مشابه في لغة البرمجة حيث يتعين علينا اتخاذ بعض القرارات ويتم تنفيذ البرنامج بناءً على ذلك. في لغة البرمجة ، يتم ذلك عن طريق هياكل التحكم. ببساطة ، التحكم في التدفق في البرمجة هو تسلسل عمليات محددة. لنبدأ بمثال بسيط ، لنفترض أننا نريد أن يكون لدينا برنامج نصي يعمل في ظل ظروف مختلفة. على سبيل المثال ، إذا قمنا بقياس درجة حرارة الهواء عند 3 درجات مئوية ، فسيتم طباعة "الهواء بارد" ، ولكن إذا كانت درجة الحرارة 21 درجة مئوية ، فسيتم طباعة "الهواء ساخن". في هذه الحالة ، يتم فحص بعض الشروط ووفقًا للشرط ، يتم تنفيذ المهمة (تتم طباعة عبارة معينة).

الاوامر الشرطية

في لغات البرمجة، في معظم الأحيان في المشاريع ، يتعين علينا التحكم في تنفيذ برنامجنا. بمعنى آخر ، نريد تنفيذ بعض الأوامر فقط في حالة استيفاء الشروط المحددة. يتم إنشاء البيانات الشرطية ، المعروفة أيضًا باسم بيانات القرار ، للقيام بذلك والتصرف بناءً على ما إذا كان شرط معين صحيحًا أم خطأ. في بايثون يمكننا اتخاذ القرارات باستخدام الأوامر التالية:

- امر **if**
- امر **if-else**
- امر **elif**
- امر **if-else** المتداخلة

امر if

في أوامر التحكم ، تكون عبارة if هي أبسط أشكالها. ينفذ الشرط ويقيمها على أنها صحيحة أو خاطئة:

```
if condition:
    statement 1
    statement 2
    statement n
```

مثال:

```
In  [1]:  num = 3
          if (num < 7):
            print("Num is smaller than 7")
Out [1]:  Num is smaller than 7
In  [2]:  a = 3
          b = 2
          if (a > b):
            print("a is greater than b")
Out [2]:  a is greater than b
```

امر if-else

باستخدام تعليمة if-else ، إذا كان شرط معين صحيحًا ، فسيتم تنفيذ العبارات الموجودة داخل بلوك if ، وإذا كان الشرط خاطئًا ، فسيتم تنفيذ بلوك else:

```
if condition:
    statement 1
else:
    statement 2
```

مثال:

```
In  [1]:  a = 1
          b = 2
          if (a > b):
            print("a is greater than b")
          else:
            print("b is greater than a")
Out [1]:  b is greater than a
In  [2]:  passing_Score = 70
          my_Score = 59
          if(my_Score >= passing_Score):
            print("Congratulations! You passed the exam")
```

```
        print("You are passed in the exam")
      else:
        print("Sorry! You failed the exam")
Out [2]: Sorry! You failed the exam
```

امر elif

بمساعدة أمر elif ، يمكننا اتخاذ قرار معقد. يتحقق الأمر elif من عدة شروط واحدة تلو الأخرى وينفذ بلوك الكود هذه إذا تم استيفاء الشرط:

```
if condition-1:
     statement 1
elif condition-2:
     stetement 2
elif condition-3:
     stetement 3
     ...
else:
     statement
```

مثال:

```
In  [1]: num = -1
         if (num > 0):
           print("Number is positive")
         elif (num < 0):
           print("Number is negative")
         else:
           print("Number is Zero")
Out [1]: Number is negative
```

امر if-else المتداخلة

تعني عبارة if-else المتداخلة أن امر if أو if-else موجودة داخل بلوك if أو if-else أخرى. وهذا بدوره يساعدنا على فحص حالات متعددة في برنامج معين.

```
if conditon_outer:
    if condition_inner:
        statement of inner if
    else:
        statement of inner else:
    statement ot outer if
else:
    Outer else
statement outside if block
```

مثال:

```
In  [1]:  num = 0
          if (num != 0):
            if (num > 0):
              print("Number is positive")
            else:
              print("Number is negative")
          else:
            print("Number is Zero")
Out [1]:  Number is Zero
```

الحلقات التكرارية

بصفتنا مبرمجين ، فإن أحد أهدافنا العامة هو كتابة تعليمات برمجية فعالة. يجب أن يدور كل شيء نقوم به حول توفير تجربة مستخدم جيدة ، وتقليل موارد وحدة المعالجة المركزية ، وإنشاء برامج بأقل قدر ممكن من التعليمات البرمجية. إحدى الطرق التي يمكننا من خلالها تحقيق ذلك هي استخدام الحلقات ، والتي يوجد منها نوعان في بايثون. تسمح لنا الحلقات بتكرار جزء من الكود متى أردنا ؛ طالما تم استيفاء الشرط الذي حددناه. تساعدنا الحلقات في تقليل تكرار الكود الخاص بنا ، حيث يتيح لنا تشغيل عملية عدة مرات.

حلقة for

باستخدام حلقة for ، يمكن تمرير أي تسلسل تكراري أو متغير. يمكن أن يكون التسلسل عبارة عن سلسلة أو قائمة أو قاموس أو صفوف أو مجموعة. كيفية استخدام حلقة for للتكرار والتنقل على النحو التالي:

```
for iterator_var in sequence:
    statements(s)
```

مثال:

```
In  [1]:  n = 4
          for i in range(n):
              print(i)
Out [1]:    0
            1
            2
            3
In  [2]:  Str = 'Persian'
          for i in Str:
              print(i)
Out [2]:  P
          e
          r
```

```
         s
         i
         a
         n
In  [3]: l = ["machine", "learning", "and"]
         for i in l:
             print(i)
Out [3]: machine
         learning
         and
```

حلقة while

في بايثون ، تُستخدم حلقة while لتنفيذ مجموعة من الأوامر بشكل متكرر حتى يتم استيفاء شرط معين ، وعندما يكون الشرط خاطئًا ، يتم تنفيذ السطر مباشرة بعد حلقة البرنامج. كيفية استخدام حلقة while هو كما يلي:

```
while expression:
    statement(s)
```

مثال:

```
In  [1]: num = 10
         sum = 0
         i = 1
         while i <= num:
             sum = sum + i
             i = i + 1
         print("Sum of first 10 number is:", sum)
Out [1]: Sum of first 10 number is: 55
In  [2]: count = 0
         while (count < 3):
             count = count + 1
             print("Hello")
Out [2]: Hello
         Hello
         Hello
```

الدوال

تُستخدم الدوال (Functions) في البرمجة لمجموعة من التعليمات التي تريد استخدامها بشكل متكرر ، أو بسبب تعقيدها ، فمن الأفضل وضعها في روتين فرعي آخر والاتصال بها عند الضرورة. تعتبر الدوال جزءًا مهمًا من أي لغة برمجة لسببين. أولاً ، يسمحون لك بإعادة استخدام

الكود الذي كتبته. على سبيل المثال ، إذا كنت تعمل باستخدام قاعدة بيانات ، فيجب عليك دائمًا التواصل مع قاعدة البيانات وإعلامها بالجدول الذي تريد الوصول إليه. عن طريق كتابة دالة ، يمكنك القيام بذلك عن طريق كتابة سطر من التعليمات البرمجية في أي برنامج يحتاج إلى الوصول إلى قاعدة البيانات. ميزة أخرى لاستخدام دالة للقيام بذلك هي أنه إذا كنت بحاجة إلى تغيير نوع قاعدة البيانات التي تستخدمها ، أو إذا وجدت عيبًا في المنطق الذي كتبت فيه الدالة لأول مرة ، فيمكنك ببساطة عمل نسخة واحدة من الدالة ويمكن للتطبيقات الأخرى استخدام النسخة المعدلة ليتم تحديثها على الفور.

السبب الثاني لاستخدام الدوال هو أنه يسمح لك بالفصل المنطقي بين المهام الفرعية المختلفة التي لديك عند العمل على أحد التطبيقات والتي تحتاج دائمًا إلى كتابتها. في مثال قاعدة البيانات ، تحتاج عمومًا إلى الاتصال بقاعدة البيانات ثم البحث في قاعدة البيانات أو إجراء بعض التغييرات. من خلال كتابة دالة للاتصال ، والثانية للاستعلام والثالثة للتحديث ، يمكنك كتابة الجزء الرئيسي من برنامجك باختصار شديد. يصبح تصحيح مثل هذا البرنامج أسهل بكثير. لأنه بمجرد تطوير مجموعة من الدوال واختبارها ، ليس من الصعب معرفة ما إذا كانت المشكلة تتعلق بإحدى الدوال أو في الكود الذي يستدعيها.

تساعد الدوال في تقسيم برنامجنا إلى أجزاء أصغر. نظرًا لأن تطبيقنا ينمو بشكل أكبر وأكبر ، فإن الدوال تجعله أكثر تنظيماً وقابلية للإدارة. بالإضافة إلى ذلك ، يمنع التكرار. يمكن أن تكون الدوال داخلية ومعرفة من قبل المستخدم.

تعريف الدالة

فيما يلي أربع خطوات لتعريف دالة في بايثون:

1. استخدم الكلمة الأساسية def للإعلان عن الدالة ، ثم اختر اسمًا للدالة.
2. أضف معاملات إلى الدالة. ضعهم بين قوسين وقم بإنهاء السطر بنقطتين (:).
3. أضف التعبيرات التي يجب أن تنفذها الدالة.
4. إذا كان يجب أن تعرض الدالة شيئًا ما في الإخراج ، فقم بإنهاء الدالة بعبارة return. بدون عبارة return ، تقوم الدالة بإرجاع كائن None.

يمكن تطبيق الخطوات اعلاه في بايثون كما يلي:

```
def function_name(parameters):
    statement(s)
    return expression
```

مثال:

```
In [1]:  def greet(name):
             print("Hello, " + name + ". Good morning!")
In [2]:  greet("ali")
```

```
Out [2]:   Hello, ali. Good morning!
In  [3]:   def absolute_value(num):
              if num >= 0:
                 return num
              else:
                 return -num
In  [3]:   absolute_value(5)
Out [3]:    5
In  [4]:   absolute_value(-8)
Out [4]:    8
In  [5]:   def evenOdd(x):
              if (x % 2 == 0):
                 print("even")
              else:
                 print("odd")
In  [6]:   evenOdd(5)
Out [6]:   odd
In  [7]:   evenOdd(8)
Out [7]:   even
```

العمل مع مكتبة NumPy

NumPy هي مكتبة بايثون تستخدم للعمل مع المصفوفات. السبب في أهمية علم البيانات باستخدام بايثون هو أن معظم المكتبات في التعلم الآلي والتعلم العميق تعتمد على NumPy كأحد كتلها الأساسية ، لأن السرعة والموارد مهمة جداً بالنسبة لها. قد تتساءل عن سبب استخدامنا لمصفوفات NumPy عند وجود قوائم بايثون. في قائمة بايثون ، يُعرض الغرض من المصفوفات. ومع ذلك ، فهي بطيئة في المعالجة ، وبطءها يكمن في كيفية تخزين الكائن في الذاكرة. كائن بايثون هو في الواقع مؤشر إلى موقع ذاكرة يخزن كل تفاصيل كائن ، مثل البايت وقيمته. في حين أن هذه المعلومات الإضافية هي ما يجعل بايثون لغة ديناميكية ، إلا أنها تأتي أيضًا بتكلفة. للتغلب على هذه المشكلة ، نستخدم مصفوفات NumPy التي تحتوي على عناصر متجانسة فقط ، أي العناصر التي لها نفس نوع البيانات. هذا يجعل تخزين المصفوفات ومعالجتها أكثر كفاءة. يهدف NumPy إلى توفير كائن مصفوفة أسرع بما يصل إلى 50 مرة من قوائم بايثون التقليدية. على عكس القوائم ، يتم تخزين مصفوفات NumPy في مكان مستمر في الذاكرة ، بحيث يمكن للعمليات الوصول إليها والتعامل معها بشكل فعال. يسمى هذا السلوك الإحالة المحلية[1] في علوم الكمبيوتر. هذا هو السبب الرئيسي وراء كون NumPy أسرع من

[1] locality of reference

القوائم. أيضًا ، يمكن إجراء العمليات الأولية باستخدام مصفوفات NumPy ، وهو أمر غير ممكن مع قوائم بايثون! هذا هو سبب تفضيل مصفوفات NumPy على قوائم بايثون عند إجراء عمليات حسابية على كميات كبيرة من البيانات.

NumPy يدير مجموعات البيانات الكبيرة بكفاءة وفعالية. بصفتك عالم بيانات أو كخبير في علم البيانات ، تحتاج إلى فهم شامل لـ NumPy وكيف يعمل في بايثون.

استيراد NumPy

عندما تريد استخدام حزمة أو مكتبةفي التعليمات البرمجية الخاصة بك، يجب عليك أولاً إتاحتها. لبدء استخدام NumPy وجميع الدوالفي NumPy، يجب عليك استيرادها. يمكن القيام بذلك بسهولة باستخدام أمر الاستيراد:

```
In [1]:   import numpy as np
```

يرمز np إلى NumPy الذي يستخدمه مجتمع علم البيانات. نقوم بتقصير NumPy إلى np لتوفير الوقت والحفاظ أيضًا على الكود القياسي بحيث يمكن لأي شخص يعمل باستخدام الكود الخاص بنا فهمه وتنفيذه بسهولة.

انشاء مصفوفة NumPy

تُستخدم طريقة np.array() لإنشاء مصفوفة أساسيةفي NumPy. الشيء الوحيد الذي يجب تضمينه هو قيم المصفوفة كقائمة:

```
In [1]:   np.array([1,2,3,4])

Out [1]:  array([1, 2, 3, 4])
```

تحتوي هذه المصفوفة على قيم عدد صحيح. يمكنك تحديد نوع البياناتفي وسيطة dtype:

```
In [2]:   np.array([1,2,3,4],dtype=np.float32)

Out [2]:  array([1., 2., 3., 4.], dtype=float32)
```

باستخدام الأقواس المربعة ([]) يمكننا الوصول إلى عناصر المصفوفة. عند الوصول إلى العناصر، ضعفي اعتبارك أن الفهرسةفي NumPy تبدأ من 0. هذا يعني أنه إذا كنت تريد الوصول إلى العنصر الأولفي المصفوفة، فيمكنك الوصول إليه باستخدام 0:

```
In [3]:   a= np.array([5 , 1, 3, 7])
          a[0]
Out [3]:   5
```

يمكن أيضًا أن تكون مصفوفات NumPy متعددة الأبعاد:

```
In  [4]:  a = np.array([[1 , 5, 2], [6, 8, 1], [10, 3, 11]])
          a
Out [4]:  array([[ 1,  5,  2],
                 [ 6,  8,  1],
                 [10,  3, 11]])
In  [5]:  a[0]
Out [5]:  array([1, 5, 2])
In  [6]:  a[2]
Out [6]:  array([10,  3, 11])
In  [7]:  a[0][0]
Out [7]:   1
```

المصفوفة الصفرية

يتيح لك NumPy إنشاء مصفوفة من الأصفار باستخدام طريقة np.zeros(). كل ما عليك فعله هو إدخال شكل[1] المصفوفة المطلوب:

```
In  [1]:  np.zeros(7)
Out [1]:  array([0., 0., 0., 0., 0., 0., 0.])
```

المصفوفة السابقة عبارة عن مصفوفة ذات بعد واحد. لإنشاء مصفوفة ثنائية الأبعاد ، قم بما يلي:

```
In  [2]:  np.zeros((2,6))
Out [2]:  array([[0., 0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0., 0.]])
```

المصفوفة الواحدية

يمكنك أيضًا إنشاء مصفوفة من الواحدات باستخدام طريقة np.ones():

```
In  [1]:  np.ones(6)
Out [1]:  array([1., 1., 1., 1., 1., 1.])
```

إضافة وحذف وفرز العناصر

يمكنك إضافة عناصر إلى المصفوفة الخاصة بك باستخدام طريقة np.append():

```
In  [1]:  a = np.array([5, 1, 2, 3, 9, 4, 7])
```

[1] shape

```
         a
Out [1]: array([5, 1, 2, 3, 9, 4, 7])
In  [2]: np.append(a, [12,2,1])
Out [2]: array([ 5,  1,  2,  3,  9,  4,  7, 12,  2,  1])
```

يتم استخدام طريقة np.delete() لحذف عنصرفي موقع معين:

```
In  [3]: a = np.array([5, 1, 2, 3, 9, 4, 7])
         a
Out [3]: array([5, 1, 2, 3, 9, 4, 7])
In  [4]: np.delete(a, 0)
Out [4]: array([1, 2, 3, 9, 4, 7])
```

بالنسبة لأي مبرمج، يعد التعقيد الزمني لكل خوارزمية مهمًا جداً. يعتبر الفرز عملية مهمة وأساسية للغاية قد تستخدمها كعالم بيانات يوميًا. لذلك، من المهم استخدام خوارزمية فرز جيدة بأقل قدر من التعقيد الزمني. تحتوي مكتبة NumPy على مجموعة واسعة من دوال الفرز التي يمكنك استخدامها لفرز عناصر المصفوفة الخاصة بك:

```
In  [5]: a = np.array([5,4,2,5,3,6,8,7,9,1,8])
         np.sort(a, kind='quicksort')
Out [5]: array([1, 2, 3, 4, 5, 5, 6, 7, 8, 8, 9])
In  [6]: a = np.array([[8,5,7,4,1,6],
                       [9,2,3,7,5,1]])
         np.sort(a, kind='mergresort')
Out [6]: array([[1, 4, 5, 6, 7, 8],
                [1, 2, 3, 5, 7, 9]])
```

تحديد شكل وحجم المصفوفة

باستخدام ndim، يمكن الحصول على عدد محاور أو أبعاد المصفوفة:

```
In  [1]: a = np.array([[8,5,7,4,1,6],
                       [9,2,3,7,5,1]])
         a.ndim
Out [1]:  2
```

size يخبرك بالعدد الإجمالي لعناصر المصفوفة:

```
In  [2]: a = np.array([[8,5,7,4,1,6],
                       [9,2,3,7,5,1]])
         a.size
```

```
Out [2]:    12
```

يستخدم shape لإيجاد شكل المصفوفة:

```
In  [3]:  a = np.array([[8,5,7,4,1,6],
                        [9,2,3,7,5,1]])
          a.shape
Out [3]:  (2, 6)
```

خلاصة الفصل الثاني

- بايثون هي واحدة من أكثر اللغات قيمة وإثارة للاهتمام لتحليل البيانات.
- بايثون هي واحدة من أسهل اللغات للبدء بها. أيضا ، هذه البساطة لا تحد من التسهيلات التي تحتاجها.
- Jupiter Notebook هو أداة قوية بشكل لا يصدق لتطوير وتقديم مشاريع علم البيانات التفاعلية التي يمكن أن تتضمن نصًا أو صورة أو صوتًا أو فيديو بالإضافة إلى تنفيذ التعليمات البرمجية.
- NumPy هي مكتبة بايثون تستخدم للعمل مع المصفوفات.
- يهدف NumPy إلى توفير كائن مصفوفة أسرع بما يصل إلى 50 مرة من قوائم بايثون التقليدية.

مصادر اضافية لمزيد من القراءة

Blum, A., Hopcroft, J., & Kannan, R. (2020). Foundations of data science. Cambridge University Press.

Godsey, B. (2017). Think Like a Data Scientist: Tackle the data science process step-by-step. Simon and Schuster.

Boschetti, A., & Massaron, L. (2016). Python data science essentials. Packt Publishing Ltd.

Deitel, P. J., & Dietal, H. (2020). Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and the Cloud. Pearson Education, Incorporated.

Zhang, N. (2020). A Tour of Data Science: Learn R and Python in Parallel. CRC Press.

McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".

Kazil, J., & Jarmul, K. (2016). Data wrangling with Python: tips and tools to make your life easier. " O'Reilly Media, Inc.".

Navlani, A., Fandango, A., & Idris, I. (2021). Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python. Packt Publishing Ltd.

Galea, A. (2018). Applied Data Science with Python and Jupyter: Use powerful industry-standard tools to unlock new, actionable insights from your data. Packt Publishing Ltd.

3

البيانات

الأهداف:

- التعرف على أنواع البيانات.
- تحضير وتنقية البيانات.
- التعرف على تقنيات تجريف الويب.
- كيفية إدخال البيانات بصيغ مختلفة.
- التمثيل المرئي للبيانات.

البيانات

يدور علم البيانات حول تجربة البيانات الخام أو المنظمة. البيانات هي حافز يمكنه توجيه الأعمال في الاتجاه الصحيح ، أو على الأقل تقديم رؤى ثاقبة يمكنها بسهولة تنظيم عمليات إطلاق منتج جديد أو اختبار تجارب مختلفة. كل هذه العناصر لها عنصر تحفيزي مشترك ، وهي البيانات. نحن ندخل العصر الرقمي حيث ننتج الكثير من البيانات. عندما تكون هذه البيانات مهمة جداً في حياتنا ، فمن المهم تخزين هذه البيانات ومعالجتها بشكل صحيح دون أخطاء. عند التعامل مع مجموعات البيانات ، يلعب نوع البيانات دورًا مهمًا في تحديد استراتيجية المعالجة المسبقة التي تعمل لمجموعة معينة للحصول على النتائج المناسبة أو نوع التحليل الإحصائي الذي يجب استخدامه للحصول على أفضل النتائج. يتيح لك فهم أنواع البيانات المختلفة اختيار نوع البيانات الذي يناسب احتياجاتك وأهدافك. سواء كنت رجل أعمال أو مسوقًا أو عالم بيانات أو محترفًا آخر يعمل مع أنواع البيانات ، يجب أن تكون على دراية بالقائمة الرئيسية لأنواع البيانات.

البيانات النوعية[1]

تصف البيانات النوعية ، أو بيانات الفئوية[2] ، كائنًا باستخدام مجموعة محدودة من الفئات المنفصلة. وهذا يعني أن هذا النوع من البيانات لا يمكن عده أو قياسه بسهولة باستخدام الأرقام وبالتالي فهو مقسم إلى فئات. جنس الشخص (ذكر أو أنثى أو غيرهم) هو مثال جيد على هذا النوع. يمكن أن تؤدي البيانات النوعية إلى الاجابة على أسئلة مثل "كيف حدث هذا؟" أو "لماذا حدث هذا؟". يعد جنس الشخص وألوانه وأعراقه أمثلة على هذا النوع من البيانات. على سبيل المثال ، تخيل تلميذاً يقرأ فقرة من كتاب في جلسة صفية. يعطي المعلم الذي يستمع إلى الكتاب ملاحظات حول كيفية قراءة الطفل لتلك الفقرة. إذا قدم المعلم ملاحظات للطفل على أساس البلاغة والنبرة والنطق دون إعطاء درجة ، فإن هذا يعتبر مثالاً على البيانات النوعية. مثال آخر يمكن أن يكون علامة تجارية للهواتف الذكية توفر معلومات حول الترتيب الحالي ولون الهاتف وفئة الهاتف وما إلى ذلك. يمكن تصنيف كل هذه المعلومات على أنها بيانات نوعية. هناك نوعان عامان من البيانات النوعية: **البيانات الاسمية[3] والمتسلسلة[4]**.

---

[1] Qualitative Data

[2] Categorical Data

[3] Nominal

[4] Ordinal

البيانات الاسمية

يتم تعريف البيانات الاسمية على أنها بيانات تُستخدم لتسمية المتغيرات أو عنونتها بدون أي كمية. عادة لا يوجد ترتيب جوهري للبيانات الاسمية. على سبيل المثال ، يمكن اعتبار لون الهاتف الذكي نوع بيانات اسمي. لأننا لا نستطيع مقارنة لون واحد مع الألوان الأخرى. بمعنى آخر ، لا يمكن القول أن "الأحمر" أكبر من "الأزرق". كمثال آخر ، لون العين هو متغير اسمي له عدة فئات (أزرق ، أخضر ، بني) ولا توجد طريقة لترتيب هذه الفئات من الأعلى إلى الأدنى.

البيانات المتسلسلة

البيانات المتسلسلة هي نوع من البيانات المصنفة بترتيب طبيعي. متغيرات البيانات المتسلسلة مدرجة بالترتيب. عادة ما يتم ترقيم المتغيرات التسلسلية للإشارة إلى ترتيب القائمة. ومع ذلك ، لا يتم قياس الأرقام أو تحديدها رياضيًا ، ولكن فقط يتم تعيينها كعنوان تعليق. على سبيل المثال، إذا أخذنا في الاعتبار حجم إحدى العلامات التجارية للملابس ، فيمكننا تصنيفها بسهولة إلى صغيرة ومتوسطة وكبيرة ، على التوالي ، بناءً على علامات الأسماء الخاصة بها.

تساعدنا هذه التصنيفات على تحديد استراتيجية الترميز التي يمكن تطبيقها على نوع البيانات. يعد ترميز البيانات مهمًا للبيانات النوعية، لأن نماذج التعلم الآلي لا يمكنها استخدام هذه القيم بشكل مباشر وتحتاج إلى تحويلها إلى أنواع رقمية لأن النماذج ذات طبيعة رياضية. بالنسبة لأنواع البيانات الاسمية حيث لا توجد مقارنة بين الفئات، يمكن استخدام ترميز one-hot، وبالنسبة لأنواع البيانات المتسلسلة، يمكن استخدام ترميز label، وهو شكل من أشكال العدد الصحيح.

البيانات الكمية[1]

البيانات الكمية هي بيانات قابلة للقياس. بمعنى آخر ، يمكن حسابها أو قياسها ويمكن الحصول على قيمة عددية لها. سعر الهاتف الذكي ، والخصم المعروض ، وتردد معالج الهاتف الذكي أو ذاكرة الوصول العشوائي لهذا الهاتف ، كلها تندرج في فئة أنواع البيانات الصغيرة. خلاصة القول هي أن هناك عددًا لا حصر له من القيم التي يمكن أن تحتوي عليها السمة. على سبيل المثال ، يمكن أن يختلف سعر الهاتف الذكي من قيمة x إلى أي قيمة. البيانات المتقطعة والمستمرة نوعان رئيسيان من البيانات الكمية.

[1] Quantitative Data

البيانات المتقطعة[1]

البيانات المتقطعة قابلة للعد وتحتوي على أعداد صحيحة فقط. عدد مكبرات الصوت المحمولة، وعدد الكاميرات ، وعدد مراكز المعالج ، وعدد بطاقات SIM المدعومة كلها أمثلة على أنواع البيانات المنفصلة.

البيانات المستمرة[2]

البيانات المستمرة هي البيانات التي يمكن تقسيمها بشكل كبير إلى مستويات أكثر دقة. يمكن قياسه على مقياس[3] أو بشكل مستمر[4] ويمكن أن يكون له أي قيمة عددية تقريبًا. على سبيل المثال، يمكنك قياس طولك بمقاييس دقيقة للغاية ، مثل الأمتار ، والسنتيمتر ، والمليمترات ، وما إلى ذلك. يمكنك تسجيل البيانات المستمرة في قياسات مختلفة لخط العرض ودرجة الحرارة والوقت وما إلى ذلك. هذا هو المكان الذي يلعب فيه الاختلاف الرئيسي مع أنواع البيانات المتقطعة. يمكن أن تأخذ المتغيرات المستمرة أي قيمة بين رقمين. على سبيل المثال ، بين 60 و 80 سم ، هناك ملايين أخرى. من القواعد الأساسية الجيدة لتحديد ما إذا كانت البيانات مستمرة أو متقطعة هي أن البيانات مستمرة إذا كان من الممكن خفض نقطة القياس إلى النصف ولا يزال الرقم الذي تم الحصول عليه معقولاً.

بيانات السلاسل الزمنية [5]

سواء كنا نريد التنبؤ باتجاهات السوق المالية أو استهلاك الكهرباء ، فإن الوقت عامل مهم يجب الآن مراعاته في نماذجنا. على سبيل المثال ، سيكون من المثير للاهتمام توقع وقت ذروة استهلاك الكهرباء في اليوم. للقيام بذلك ، يكفي استخدام بيانات السلاسل الزمنية. بيانات السلاسل الزمنية هي سلسلة من الأرقام التي يتم جمعها على فترات منتظمة خلال فترة زمنية. في السلسلة الزمنية، غالبًا ما يكون الوقت متغيرًا مستقلاً والهدف عادةً هو عمل تنبؤات للمستقبل.

تحليل السلاسل الزمنية هو طريقة خاصة للتحليل المتسلسل لنقاط البيانات التي تم جمعها خلال فترة زمنية. في تحليل السلاسل الزمنية ، يسجل المحللون نقاط البيانات على فترات منتظمة خلال فترة زمنية ، بدلاً من مجرد تسجيل نقاط البيانات بشكل متقطع أو عشوائي. ومع ذلك ، فإن هذا النوع من التحليل ليس مجرد عملية جمع البيانات بمرور الوقت. ما يميز بيانات

---

[1] Discrete

[2] Continuous

[3] scale

[4] continuum

[5] Time Series Data

السلاسل الزمنية عن البيانات الأخرى هو أن التحليل يمكن أن يُظهر كيف تتغير المتغيرات بمرور الوقت. بمعنى آخر ، يعد الوقت متغيرًا مهمًا ، لأنه يوضح كيفية تنظيم البيانات على طول المسار بالإضافة إلى النتائج النهائية. يوفر مصدرًا للمعلومات الإضافية ومجموعة من التبعيات بين البيانات.

يتطلب تحليل السلاسل الزمنية عادةً عددًا كبيرًا من نقاط البيانات لضمان الاستقرار والموثوقية. تضمن مجموعة البيانات الكبيرة أن لديك ما يكفي من عينات الفهرس وتقليل تحليل البيانات الصاخبة. كما أنه يضمن عدم اكتشاف اتجاهات أو أنماط بعيدة المنال.

يساعد تحليل السلاسل الزمنية المؤسسات على فهم الأسباب الكامنة وراء الاتجاهات أو الأنماط النظامية[1] بمرور الوقت. باستخدام التمثيل الصوري للبيانات ، يمكن لمستخدمي الأعمال رؤية الاتجاهات الموسمية واستكشاف سبب هذه الاتجاهات بعمق أكبر. باستخدام نماذج التحليل الحديثة ، يمكن لهذه الرسوم التوضيحية أن تتجاوز الرسوم البيانية الخطية. عندما تحلل المؤسسات البيانات على فترات منتظمة ، يمكنها أيضًا استخدام التنبؤ بالسلاسل الزمنية للتنبؤ باحتمالية الأحداث المستقبلية. يعد توقع السلاسل الزمنية جزءًا من التحليل التنبئي. من المرجح أن يشير هذا إلى التغييرات في البيانات ، مثل السلوك الموسمي أو الدوري ، والتي توفر فهمًا أفضل لمتغيرات البيانات وتساعد على التنبؤ بشكل أفضل.

يستخدم تحليل السلاسل الزمنية للبيانات غير الثابتة ؛ العناصر التي تتقلب باستمرار بمرور الوقت أو تتأثر بمرور الوقت. غالبًا ما تستخدم الصناعات مثل التمويل والتجزئة والاقتصاد تحليل السلاسل الزمنية ، حيث تتغير العملات والمبيعات باستمرار. تحليل سوق الأوراق المالية هو مثال رائع على تحليل السلاسل الزمنية في الممارسة. وبالمثل ، يعد تحليل السلاسل الزمنية مثاليًا للتنبؤ بتغير المناخ ويساعد خبراء الأرصاد الجوية على توقع كل شيء بدءًا من تقرير الطقس في الغد وحتى تغير المناخ لسنوات قادمة. أمثلة على تحليل السلاسل الزمنية في الممارسة هي:

- بيانات الطقس.
- قياس هطول الأمطار.
- قياس معدل ضربات القلب[2] (EKG).
- مراقبة الدماغ[3].
- مبيعات ربع سنوية.
- سعر السهم.

---

[1] Systemic

[2] Heart rate monitoring

[3] Brain monitoring

- تداول الأسهم الآلي.
- سعر الفائدة.

مجموعة البيانات والميزات

يمكن غالبًا اعتبار مجموعة البيانات على أنها مجموعة كائنات بيانات لها نفس الخصائص. تشمل اسماء أخرى لكائن البيانات: سجل، أو نقطة، أو متجه، أو نمط، أو حدث، أو عنصر، أو عينة، أو مثال، أو عرض، أو كيان.في المقابل، يتم وصف كائنات البيانات من خلال عدد من السمات التي تمثل الخصائص الرئيسية لكائن ما، مثل وقت حدوث الحدث. على سبيل المثال، يختلف لون العين من شخص لآخر، وتختلف درجة حرارة الجسم بمرور الوقت. وتجدر الإشارة إلى أن لون العين خاصية رمزية بعدد قليل من القيم الممكنة (بني، أسود، أزرق، أخضر، إلخ)، بينما درجة الحرارة هي خاصية عددية بعدد غير محدود من القيم. الاسماء الأخرى هي متجه أو متغير أو عينة أو سمة أو بُعد.

غالبًا ما تكون مجموعة البيانات عبارة عن ملف يكون فيه كل كائن في صف وكل عمود يتوافق مع إحدى ميزات هذه الكائنات. على سبيل المثال، يوضح الجدول 3ـ1 مجموعة بيانات تحتوي على معلومات الطالب. يشير كل صف إلى طالب، وكل عمود عبارة عن ميزة تصف بعض جوانب الطالب، مثل رقم الطالب وسنة الالتحاق ومتوسط درجة الدرجات ومجال الدراسة.

جدول 3ـ1 مثال على مجموعة بيانات تحتوي على معلومات الطالب

رقم الطالب	سنة الالتحاق	المعدل	مجال الدراسة
976001	1397	18.45	هندسة حاسبات
974120	1397	19.03	علوم حاسبات
990245	1399	18.95	هندسة كهرباء

الميزات العامة لمجموعة البيانات

ثلاث خصائص عامة تُستخدم عند استخدام العديد من مجموعات البيانات ولها تأثير كبير على استخدام تقنيات التعلم الآلي هي: الأبعاد[1] والتشتت[2] والدقة[3].

- **الابعاد:** أبعاد مجموعة البيانات هي عدد الميزات التي تمتلكها الكائنات في مجموعة البيانات. تختلف البيانات منخفضة الأبعاد نوعياً عن البيانات المتوسطة أو عالية الأبعاد. في

[1] dimensionality

[2] sparsity

[3] resolution

الواقع ، أحيانًا تسمى المشكلات المتعلقة بتحليل البيانات عالية الأبعاد بالمشكلات متعددة الأبعاد. لهذا السبب ، تعتبر الأبعاد حافزًا مهمًا في معالجة البيانات.

- **التشتت:** في مجموعة بيانات ، يشير المشتت إلى عدد السجلات في جدول ليس له قيمة. بمعنى آخر ، في بعض مجموعات البيانات ، تحتوي معظم خصائص الكائن على قيم 0. من الناحية العملية ، يعد التبعثر او التشتت ميزة ، حيث يجب تخزين وترتيب القيم غير الصفرية فقط. هذا يوفر وقتًا كبيرًا في الحساب والتخزين.
- **الدقة (دقت العرض):** غالبًا ما يكون من الممكن الحصول على بيانات بمستويات مختلفة من الدقة ، وكذلك تختلف خصائص البيانات باختلاف درجات الدقة. على سبيل المثال ، يبدو سطح الأرض غير مستوٍ للغاية بدقة تصل إلى بضعة أمتار ، ولكنه مسطح نسبيًا بدقة تصل إلى عشرات الكيلومترات. تعتمد الأنماط في البيانات أيضًا على مستوى الدقة. إذا كان مستوى الدقة جيدًا جدًا ، فقد لا يكون النمط مرئيًا أو قد يتم حذفه بين الضوضاء. إذا كانت الدقة كبيرة جدًا ، فقد يختفي النمط. على سبيل المثال ، تعكس التغيرات في الضغط الجوي على مقياس الساعة حركة العواصف وأنظمة الطقس الأخرى. على نطاق عدة أشهر ، لا يمكن اكتشاف هذه الظواهر.

أمثلة على البيانات عالية الأبعاد

توضح الأمثلة التالية بيانات عالية الأبعادفي مجالات مختلف:

مثال 1: البيانات الصحية

البيانات عالية الأبعاد شائعةفي مجموعة من البيانات الصحية أن عدد خصائص شخص معين يمكن أن يكون كبيرًا جدًا، على سبيل المثال، ضغط الدم، ومعدل ضربات القلب أثناء الراحة، والحالة المناعية، والتاريخ الجراحي، والطول، والوزن، والحالة، إلخ.في مجموعة البيانات هذه، من الشائع أن يتجاوز عدد السمات عدد المشاهدات.

مثال 2: البيانات المالية

البيانات عالية الأبعاد شائعة أيضًافي مجموعات البيانات المالية، حيث يمكن أن يكون عدد خصائص سهم معين كبيرًا جدًا، على سبيل المثال، حجم التداول ونسبة PE ومعدل الأرباح وما إلى ذلك.في هذا النوع من البيانات، من الشائع أن يكون عدد السمات أكبر بكثير من عدد المشاركات الفردية.

مثال 3: المحتوى الجيني

غالبًا ما تحدث البيانات عالية الأبعادفي المحتوى الجيني، حيث يمكن أن يكون عدد سمات الجينات لفرد معين هائلاً.

كيفية إدارة البيانات عالية الأبعاد

لا يمكن تحليل الأبعاد الفردية على نطاق صغيرفي البيانات عالية الأبعاد. قد يستغرق الأمر أيامًا أو شهورًا لإجراء تحليل ذي مغزى يتطلب الكثير من الوقت والمال. سوف يسبب لنا تدريب البيانات عالية الأبعاد مشاكل:

- مع زيادة الابعاد ، تزداد المساحة المطلوبة لتخزين المعلومات.
- مع زيادة الأبعاد ، تزداد أيضًا إمكانية زيادة الضبط الزائد Overfitting.
- كلما كبرت الأبعاد ، زاد تعقيد الوقت في تدريب النموذج.
- لا يمكننا تصور حجم كبير من البيانات. من خلال تقليل الأبعاد ، نقوم بتقليل البيانات إلى ثنائية أو ثلاثية الأبعاد لتحسين التمثيل الصوري.

لذلك، للتعامل مع هذه المشاكل، من الضروري إدارة البيانات عالية الأبعاد. هناك طريقتان شائعتان للتعامل مع البيانات عالية الأبعاد:

1. **اختيار ميزات أقل**

الطريقة الأكثر وضوحًا لتجنب التعامل مع البيانات الكبيرة هي ببساطة اختيار ميزات أقل من مجموعة البيانات. هناك عدة طرق لتحديد ما إذا كنت تريد إزالة السمات من مجموعة البيانات، بمافي ذلك:

- **حذف السمات مع العديد من القيم المفقودة:** إذا كان عمود معين في مجموعة بيانات يحتوي على كميات كبيرة من البيانات المفقودة ، فقد تتمكن من حذفها بالكامل دون فقد الكثير من المعلومات.
- **حذف السمات ذات التباين المنخفض:** إذا كان عمود معين في مجموعة بيانات يحتوي على قيم لا تتغير إلا قليلاً ، فقد تتمكن من حذفها ، لأنه من غير المحتمل أن توفر معلومات مفيدة حول متغير استجابة مقارنة بالسمات الأخرى.
- **استبعاد السمات ذات الارتباط المنخفض مع متغير الاستجابة:** إذا كانت سمة معينة لا علاقة لها بمتغير الاستجابة محل الاهتمام ، فيمكنك على الأرجح إزالتها من مجموعة البيانات لأنه من غير المحتمل أن تكون سمة مفيدة في النموذج.

2. **استخراج الميزات**

تقنية أخرى شائعة لإدارة البيانات عالية الأبعاد هي استخراج الميزات. الغرض من استخراج الميزات هو تقليل عدد الميزاتفي مجموعة البيانات عن طريق إنشاء ميزات جديدة من الميزات الموجودة (ثم حذف الميزات الأصلية). يجب أن تكون مجموعة الميزات الجديدة المصغرة هذه قادرة على تلخيص معظم المعلومات الموجودةفي مجموعة الميزات الرئيسية.

بهذه الطريقة، يمكن إنشاء ملخص للميزات الرئيسية من خلال الجمع بين المجموعة الأصلية.

يتمثل الاختلاف بين اختيار الميزة واستخراج الميزة فى أن اختيار الميزة يرتب أهمية الميزات فى مجموعة البيانات ويتجاهل الميزات الأقل أهمية (لا يتم إنشاء ميزات جديدة)، بينما يؤدى استخراج الميزات إلى إنشاء ميزات جديدة من الميزات الموجودة عن طريق تقليل الميزات.

جمع البيانات[1]

لكل برنامج أهدافه ومتطلباته الخاصة التي يجب الوفاء بها. يجب استخدام استراتيجيات مختلفة لتحقيق مخرجات موثوقة في النهاية. تؤدي هذه الحاجة إلى فكرة إنشاء مجموعات بيانات جديدة يمكن استخدامها لاحقًا لمجموعة متنوعة من الأغراض. يعد إنشاء مجموعة بيانات كبيرة أمرًا شاقًا إذا تم إجراؤه يدويًا. ولكن يمكن لأساليب مثل تجريف الويب[2] و زاحف الشبكة أتمتة عملية جمع البيانات وتسهيل إنشاء مجموعات البيانات للتحليل.

تجريف ويب

في عالم اليوم التنافسي، يبحث الجميع عن طرق للابتكار واستخدام تقنيات جديدة. يوفر تجريف الويب حلاً لأولئك الذين يرغبون في الوصول تلقائيًا إلى بيانات الويب المنظمة. يعد تجريف الويب مفيدًا إذا كان موقع الويب العام الذي تريد تلقي المعلومات منه لا يحتوي على واجهة برمجة تطبيقات أو لديه وصول محدود فقط إلى البيانات.

بشكل عام، يتم استخدام التنقيب عن بيانات الويب من قبل الأفراد والشركات الذين يرغبون في استخدام الكمية الهائلة من بيانات الويب المتاحة للجمهور لاتخاذ قرارات ذكية. يمكن القيام بهذه الطريقة يدويًا باستخدام طرق إدراج النسخ[3] التقليدية، ولكنفي معظم الحالات يُفضل استخدام الأدوات الآلية لأنها أقل تكلفة وتعمل بشكل أسرع. يمكن تقسيم عملية تجريف الويب بأكملها إلى مراحل مختلفة وشرحها بإيجاز على النحو التالي:

---

[1] Data collection

[2] web scraping

[3] copy-pasting

- **المرحلة الأولى — جلب البيانات**[1]: في هذه الخطوة ، يجب تحديد مواقع الويب التي يمكن من خلالها الوصول إلى البيانات. يمكن بعد ذلك إجراء الجلب باستخدام بروتوكول HTTP ، وهو بروتوكول إنترنت يُستخدم لإرسال الطلبات وتلقيها من خادم الويب.
- **المرحلة الثانية ـ استخراج المعلومات**[2]: بعد جلب مستندات HTML المطلوبة ، فإن الخطوة التالية هي استخراج المعلومات التي نحتاجها من موقع الويب. يمكن القيام بذلك باستخدام عدة تقنيات مثل تحليل HTML و DOM و XPath ومطابقة أنماط النص.
- **المرحلة الثالثةـ تحويل البيانات**[3]: بعد استخراج المعلومات المطلوبة من المواقع المطلوبة (URL) ، ستكون البيانات غير منظمة. يمكن بعد ذلك تحويلها إلى نموذج منظم مثل CSV أو جدول بيانات أو pdf ، للعرض التقديمي أو التخزين.

تقنيات تجريف الويب

هناك العديد من الطرق لتطبيق تجريف الويب ، والتي يمكن من خلالها اختيار أفضل طريقة بناءً على احتياجات المبرمج. وفقًا لما سبق ، يمكن ملاحظة أن هذه التقنيات تُستخدم في مرحلة الاستخراج وتنقسم بشكل أساسي إلى فئتين: تجريف الويب اليدوي وتجريف الويب التلقائي. يناقش هذا القسم بعض التقنيات الرئيسية في كل فئة ويوفر أساسًا لاختيار التقنية المناسبة فيما بينها.

تجريف الويب اليدوي

ليس من المستغرب أن يتم تجريف الويب يدويًا. يُعد إدراج النص التقليدي طريقة يدوية يتم فيها نسخ البيانات المراد استخراجها من موقع ويب يدويًا كمجموعة وإدراجهافي مستند. ثم يتم جمع البيانات المطلوبة من المجموعة وترتيبها بطريقة منظمة.في بعض الأحيان، للحصول على معلومات أقل، يمكن أن يكون هذا هو أفضل أسلوب. ولكنفي عملية إنشاء مجموعة كبيرة من البيانات، يمكن أن تكون هذه التقنية مملة وعرضة للخطأ، لأنها تنطوي على الكثير من العمل اليدوي. هذه الطريقة لها العديد من المزايا والعيوب:

- **المزايا**
  - ← يمكن القول إنها أسهل طريقة لتجريف الويب ، حيث أنك لست بحاجة إلى تعلم مهارات جديدة للقيام بذلك.
  - ← افراد يسمح للأشخاص بمسح أي نقطة من البيانات وتجنب الأخطاء أو حذف البيانات غير ذات الصلة أثناء الاستخراج.

---

[1] Fetch data

[2] Extracting Information

[3] Data Transformation

← نظرًا لبطء سرعة تجريب الويب اليدوي ، فمن غير المحتمل أن يتم حظر الوصول إلى موقع الويب الذي تستخرج البيانات منه.

▪ **العيوب**

← إنها بالتأكيد أبطأ طريقة لتجريف الويب. حتى بأقصى سرعة ، يكون الروبوت المستند إلى تجريف الويب أسرع في استخراج البيانات من الإنسان.

← يؤدي العمل البشري أيضًا إلى خطأ بشري. اعتمادًا على أهمية البيانات ، يمكن أن يكلفك الخطأ البشري غالياً.

تجريف الويب التلقائي

على الجانب الآخر من تجريف الويب اليدوي يدوية، هناك تجريف ويب تلقائي. تجريف ويب تلقائي مشهور بشكل متزايد بسبب سهولة استخدامه وتوفير الوقت والتكلفة. يتم تنفيذ هذا النهج من خلال تقنيات مختلفة، والتي سنصفها فيما يلي. هذه الطريقة لها أيضًا العديد من المزايا والعيوب:

▪ **المزايا**

← بالنسبة لتجريف الويب ، فهو يعمل بسرعة مذهلة ، ويستخرج مئات السجلات في ثوانٍ.

← من السهل استخدامه. يمتلك معظم مجرفي الويب المعاصرين واجهات مستخدم مدمجة شديدة البساطة تسمح لأي شخص باستخراج البيانات من الويب دون الحاجة إلى مهارات البرمجة.

▪ **العيوب**

← قد يحتاجون إلى بعض التدريب التفصيلي حول كيفية استخدام الأداة نفسها. تعالج بعض الأدوات هذه المشكلة من خلال تنفيذ واجهات مستخدم بسيطة.

← تحاول بعض مواقع الويب بنشاط منع صفحات الويب من تجريف الويب الخاصة بهم.

← عدم وجود تحكم بشري أثناء استخراج البيانات. يوصى بالتحقق من البيانات قبل استخدام مجموعة البيانات المستخرجة من مجرفي الويب التلقائيين.

تحليل HTTP

بشكل عام، تتم عملية تحليل[1] سلسلة الرموزفي اللغة الطبيعية أو لغة الكمبيوتر أو بنية البيانات وفقًا للقواعد النحوية. عادة ما تكون نتيجة تحليل المستند عبارة عن شجرة بها مجموعة من العقد التي تمثل هيكلها.في تحليل HTML، بعد جلب مستند HTML، يتم إنشاء شجرة من العقد

---

[1] parsing

أثناء التحليل، والتي يمكن من خلالها استخلاص معلومات مثل عناوين الصفحات والعناوين وفقرات الصفحة عن طريق التعرف على عُقد HTML. يمكن استخدام بعض لغات الاستعلام المهيكلة، مثل XQuery و HTQL، لتحليل صفحات HTML واسترداد محتوى الصفحة وتعديله.

تحليل DOM

يعد DOM معمارية[1] أساسية شائعة لإدارة هيكل المستندات التي تعمل على مستندات XML من خلال إنشاء واجهة للوصول إلى هيكل هذه المستندات ومحتوياتها. مثل محللات HTML، عندما يتم جلب مستند XML وتطبيقه باستخدام محلل DOM، يتم تشكيل بنية شجرة تحتوي على جميع عناصر المستند. بمساعدة DOM، يمكن فحص محتويات وهيكل الوثيقة واستخدامها للاستخراج.

تحليل XPath

XPath تعني لغة مسار[2] XML. يمكن استخدام هذه التقنيةفي مستندات XML للوصول إلى العناصر المختلفةفي هيكلها ومحتواها. يمكن أيضًا استخدام XPath للوصول إلى مستندات HTML، حيث أن لها بنية مشابهة لـ XML. يتيح لنا XPath كتابة عبارات يمكنها الوصول مباشرة إلى عناصر HTML بدلاً من فحص الشجرة بأكملها. بشكل عام، بعد تحليل DOM، يمكن استخدام XPath كمجرف ويب لاستخراج البيانات. XPath ليست لغة ، إنها تأتيفي شكل عبارات يجب أن تكون محددة.

محددات CSS

تقنية تجريف الويب الشائعة الأخرى لاستخراج البيانات من مستندات HTML هي استخدام محددات CSS[3]. CSS هي لغة تستخدم لدمج مستندات HTML وتصف بشكل أساسي عرض المستندات المهيكلة مثل HTML و XML. بناءً على خصائص CSS المختلفة، يتم استخدام محددات مختلفة مثل النوع والسمة والمعرف وما إلى ذلك للإشارة إلى بُنية ومحتوى موقع الويب. يمكن استخدام هذه العناصر لمطابقة واستخراج عناصر HTML.

مطابقة نمط النص

مطابقة نمط النص هي تقنية مطابقة[4] تستخدم التعابير النمطية[5] لمطابقة علامات HTML واستخراج البيانات من مستندات HTML. التعابير النمطية هي بشكل عام سلسلة من الحروف

---

[1] platform

[2] XML Path Language

[3] Cascading Style Sheets

[4] matching technique

[5] Regular expressions

تؤدي إلى نمط بحث. نظرًا لأن HTML يتكون من العديد من السلاسل تقريبًا، يمكن تطبيق التعابير النمطية هنا عن طريق مطابقة سلاسل مختلفة. لكن التعابير النمطية قد لا تكون الخيار الأولفي تحليل HTML، حيث توجد فرصة لارتكاب أخطاء مثل العلامات المفقودة.

مكتبات تجريف الويب

توجد مجموعة ضخمة من مكتبات البايثون لتجريف الويب. ولكن أيهما يجب أن يتم اختياره لمشروع معين؟ أي من هذه المكتبات تتمتع بأكبر قدر من المرونة؟ الغرض من هذا القسم هو الإجابة على هذه الأسئلة من خلال مراجعة العديد من مكتبات بايثون الشائعة لتجريف الويب التي يجب أن يكون أي شخص مهتم على علم بها.

Requests

Requests هي مكتبة بايثون الأساسية لتجريف الويب. يمكنه إرسال طلبات HTML إلى خادم الويب لاسترداد البيانات من صفحة الويب. تُستخدم هذه المكتبة عملية تجريف الويبفي مرحلة إحضار المعلومات. تمنح مكتبة بايثون هذه المستخدمين درجة عالية من المرونة من خلال توفير أنواع مختلفة من طلبات HTTP مثل GET وPOST. نظرًا لأن هذه مكتبة أساسية لا يمكن استخدامها إلا لجلب صفحات الويب، فلا يمكن استخدامها بشكل منفصل لجمع البيانات. لذلك، يجب دمجه مع مكتبات أخرى للحصول على مخرجات موثوقة.

كيف يمكننا استخدام هذه المكتبة؟ بعد الحصول على HTML خام صفحة ويب بسيطة، فأنت بحاجة إلى تحليلها واستخراج البيانات التي تحتاجها. دعونا نلقي نظرة على مثال جرفنا فيه صفحة ويكيبيديا حول "التعلم العميق".

غالبًا ما يتم تضمين مكتبة Requestsفي مكتبات بايثون الداخلية، ولكن إذا لم تتمكن من إدخالها لسبب ما، فما عليك سوى تشغيل الأمر التاليفي سطر الأوامر:

```
› pip install requests
```

بعد تثبيت المكتبة، نحتاج إلى استيرادها إلى مشروعنا. بعد ذلك، نحتاج إلى إرسال طلب GET إلى عنوان URL:

```
In [1]: import requests
        r =requests.get('https://fa.wikipedia.org/wiki/التعلم_العميق ')
        print(r.content)
Out [1]: b'<!DOCTYPE html>\n<html class="client-nojs" lang="fa" dir="rtl">\n<head>\n<meta charset="UTF-8"/>\n<title>.......
```

LXML

كما هو مذكورفي القسم السابق، كان لـ Requests قيود لا يمكن استخدامها كمحلل. LXML هي أداة سريعة لتحليل HTML و XMLفي بايثون يمكن استخدامها لتحليل واستخراج البيانات من صفحات الويب. هذه المكتبة أسرع من معظم المحللين وتستخدم API بايثون لتسهيل استخدامها. لكنها لا تعمل بشكل جيد مع مستندات HTML سيئة التصميم. هذا يجعل LXML أقل مرونة من المكتبات الأخرى..

لتثبيت هذه المكتبة، ما عليك سوى تشغيل الأمر التاليفي سطر الأوامر:

```
> pip install lxml
```

في هذا المثال، نحاول إظهار جميع الروابط الموجودة على صفحة ويب واحدة (ولكننا نعرض بعضًا منها فقطفي الإخراج). نستخدم مكتبة Requests مرة أخرى للحصول على كود HTML الخام لصفحة الويب ثم تحليلها باستخدام LXML.

```
In [1]:  import requests
         import lxml.html
         r =requests.get('https://fa.wikipedia.org/wiki/التعلم_العميق)
         content = r.content
         doc = lxml.html.fromstring(content)
         for element in doc.xpath('//a/@href'):
           print(element)
Out [1]:
         https://da.wikipedia.org/wiki/Deep_learning
         https://de.wikipedia.org/wiki/Deep_Learning
         https://en.wikipedia.org/wiki/Deep_learning
         https://es.wikipedia.org/wiki/Aprendizaje_profundo
         https://et.wikipedia.org/wiki/S%C3%BCgav%C3%B5pe
         https://eu.wikipedia.org/wiki/Ikaskuntza_sakon
         https://fi.wikipedia.org/wiki/Syv%C3%A4oppiminen
         https://fr.wikipedia.org/wiki/Apprentissage_profond
         https://it.wikipedia.org/wiki/Apprendimento_profondo
         https://ms.wikipedia.org/wiki/Pembelajaran_dalam
         .....
```

في مقتطف الشفرة أعلاه، استخدمنا عبارة XPath لتحديد جميع الروابط التي وجدناها وطباعتها.

BeautifulSoup

BeautifulSoup هي مكتبة بايثون تُستخدم لاستخراج المعلومات من ملفات XML و HTML. تم استدعاء هذه المكتبة لأسباب جميلة لأنها تساعدك على تحليل البيانات المستخرجة

بسهولة والتمرير خلالها واختيار البيانات التي تختارها فقط. اكتسب BeautifulSoup شهرة نظرًا لسهولة استخدامه، ولكنه أبطأ من LXML. تتمثل إحدى المزايا المهمة لهذه المكتبةفي أنها مناسبة لأي نوع من مواقع الويب (لديها القدرة على التعرف على ترميز[1] الصفحة وبالتالي تلقي معلومات أكثر دقة من نص HTML) ويمكن استخدامها مع Requests لتنفيذ لخطوات الجلب والاستخراج الناجحة. يمكن دمجه مع المحلل LXML.

لتثبيت هذه المكتبة، ما عليك سوى تشغيل الأمر التاليفي سطر الأوامر:

```
› pip install bs4
```

في هذا المثال نحاول تحليل جميع فقرات المثال السابق وطباعة المحتوى (يتم عرض القليل منها هنا):

```
In  [1]:  import requests
          from bs4 import BeautifulSoup
          r =requests.get('https://fa.wikipedia.org/wiki/التعلم_العميق')
          content = r.content
          soup = BeautifulSoup(content, features="html.parser")
          for element in soup.findAll('p'):
            print(element.text)
Out [1]:
          على سبيل المثال ،في معالجة الصور ، يمكن للطبقات السفلية التعرف على الحواف ، بينما قد
          تتعرف الطبقات العليا على ميزات بشرية أكثر أهمية ، مثل الحروف أو الوجوه.

          قبل ظهور التعلم العميق ، اعتمدت أساليب التعلم الآلي التقليدية بشكل كبير على التمثيلات (اختيار
          الميزات) المستمدة من البيانات. تتطلب هذه الأساليب خبيرًافي مجال الموضوع لإجراء استخراج
          الميزات يدويًا. ومع ذلك ، يعد استخراج الميزات يدويًا عملية صعبة وتستغرق وقتًا طويلاً.كان
          ظهور التعلم العميق قادرًا على استبدال هذه الأساليب التقليدية بسرعة. لأنه يمكنه استخراج
          الخصائص تلقائيًا لتناسب أي مشكلة
```

في مقتطف الشفرة السابق، طلبنا من BeautifulSoup استخدام المحلل اللغوي "html.parser" للمحتوى المستخرج وتحديد جميع علامات <P> لنا.

Selenium

المكتبات المذكورة حتى الآن لها قيود لا يمكنها العمل مع مواقع الويب المصممة باستخدام JavaScript. هذا يجعل من الصعب العمل مع صفحات الويب الديناميكيةفي تجريف الويب. يعد العمل مع صفحات الويب الديناميكية أحد أكبر التحدياتفي تجريف الويب، لكن Selenium هي واحدة من مكتبات بايثون التي يمكنها التغلب على هذه المشكلة. Selenium هي أداة مفتوحة

[1] encoding

المصدر قائمة على الويب وبرنامج تشغيل ويب ، مما يعني أنه يمكنك استخدامها لفتح صفحة ويب ، والنقر فوق زر والحصول على النتائج.

على الرغم من قوتها، يعتبر Selenium أداة بدائية. كما يسمح للشفرة بتقليد السلوك البشري. ومع ذلك، فإن أحد القيود الرئيسية لهذه المكتبة هو تنزيل وتنفيذ JavaScript لكل صفحة، مما يجعل العمليات أبطأ وغير مناسبة للمشاريع الكبيرة.

لتثبيت هذه المكتبة، ما عليك سوى تشغيل الأمر التالي في سطر الأوامر:

```
> pip install selenium
```

لاحظ أن استخدام Selenium لتنزيل محتويات صفحة الويب لا يتطلب مساعدة المكتبات الأخرى، حيث يمكن Selenium أن يفعل كل شيء بمفرده! بمجرد إخبارنا لبرنامج تشغيل الويب بأننا سنستخدم chromedriver وأي عنوان URL يجب أن يجرف، نحتاج إلى تحديد ما نبحث عنه في البيانات المستخرجة. في هذا المثال، حاولنا الحصول على جميع الروابط مرة أخرى:

In [1]:
```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
import requests
from selenium import webdriver
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')
wd = webdriver.Chrome('chromedriver',chrome_options=chrome_options)
driver=webdriver.Chrome('chromedriver',chrome_options=chrome_option
s)
driver.get('https://fa.wikipedia.org/wiki/التعلم_العميق')
links = driver.find_elements_by_tag_name('a')
for element in links:
  print(element.get_attribute('href'))
```

Out [1]:
```
https://da.wikipedia.org/wiki/Deep_learning
https://de.wikipedia.org/wiki/Deep_Learning
https://en.wikipedia.org/wiki/Deep_learning
https://es.wikipedia.org/wiki/Aprendizaje_profundo
https://et.wikipedia.org/wiki/S%C%3BCgav%C%3B5pe
https://eu.wikipedia.org/wiki/Ikaskuntza_sakon
https://fi.wikipedia.org/wiki/Syv%C%3A4oppiminen
```

تخزين البيانات وعرضها

اعتمادًا على طبيعة البيانات، يمكن تخزينهافي مجموعة متنوعة من التنسيقات. تخزن بعض التنسيقات البيانات بطريقة يمكن للآلات التحكم فيها بسهولة، بينما يقوم البعض الآخر بتخزين البيانات بطريقة يمكن للبشر قراءتها. مستندات Microsoft Word هي مثال على هذا الأخير.في المقابل، CSV و JSON و XML هي أمثلة على الطريقة الاولى.في هذا القسم، سنناقش أولاً بإيجاز كل من تنسيقات تخزين البيانات هذه، ثم نفحص كيفية قراءة هذه الملفات التي يتحكم فيها الجهاز.

CSV (قيم مفصولة بفواصل [1])

أول ملف يمكن للأجهزة قراءته هو ملف CSV.في ملف CSV، يتم فصل أعمدة البيانات بفواصل. هذه طريقة لتبادل البيانات المهيكلة بين التطبيقات التي لا تحتاج بالضرورة إلى التحدث مباشرة مع بعضها البعض.

كيفية استيراد بيانات CSV في بايثون

هناك عدة طرق لقراءة ملف CSV يستخدم وحدة CSV أو مكتبة pandas:

- **وحدة csv:** وحدة CSV هي إحدى وحدات بايثون التي توفر فئات لقراءة وكتابة بيانات الجدول بتنسيق ملف CSV.
- **مكتبة pandas:** مكتبة pandas هي إحدى مكتبات بايثون مفتوحة المصدر التي توفر أدوات وتقنيات تحليل البيانات.

استخدام csv.reader()

باستخدام بضعة أسطر من التعليمات البرمجية أدناه، يمكن لـبايثون فتح ملف بيانات CSV وتحويل البيانات حتى تتمكن بايثون من فهمها. يستخدم مقتطف الكود التالي دالة csv.reader() لقراءة ملف data.csv لتعيين البيانات إلى القوائم:

```
In [1]: import csv

        # opening the CSV file
        with open('data.csv', mode ='r')as file:

          # reading the CSV file
```

[1] Comma-Separated Values

```
    csvFile = csv.reader(file)

    # displaying the contents of the CSV file
    for lines in csvFile:
        print(lines)
```

Out [1]:

```
['id', 'name', 'age', 'Field of Study']
['9396321', 'sara', '26', 'electrical engineering']
['9496328', 'mahsa', '24', 'Computer Engineering']
['9896325', 'maryam', '23', 'computer science']
['9496352', 'sepideh', '25', 'electrical engineering']
['9896389', 'sima', '26', 'computer science']
['9896325', 'mina', '21', 'electrical engineering']
```

مبدئيًا، يتم فتح ملف CSV باستخدام طريقة open() في الوضع 'r' (يحدد وضع القراءة عند فتح ملف) والذي يُعيد كائن الملف. بعد ذلك، يتم تخزين إخراج الدالة csv.reader(file) في متغير csvFile. يحتوي المتغير csvFile الآن على قارئ CSV بالملف المفتوح. يتيح لنا قارئ CSV هذا عرض بيانات ملفاتنا بسهولة باستخدام أوامر بايثون البسيطة.

ملاحظة: تُستخدم الكلمة المحجوزة "with" جنبًا إلى جنب مع طريقة open()، لأنها تبسط معالجة الاستثناءات وتغلق ملف CSV تلقائيًا.

استخدام csv.DictReader()

تشبه هذه الطريقة الطريقة السابقة ، أولاً يتم فتح ملف CSV باستخدام طريقة open()، ثم يتم تعيين البيانات الموجودة في ملف CSV إلى قاموس باستخدام كلاس DictReader من وحدة csv:

In [1]:

```
import csv

# opening the CSV file
with open('data.csv', mode ='r')as file:

    # reading the CSV file
    csvFile = csv.DictReader(file)

    # displaying the contents of the CSV file
    for lines in csvFile:
        print(lines)
```

Out [1]:

```
OrderedDict([('id', '9396321'), ('name', 'sara'), ('age', '26'), ('Field of Study', 'electrical engineering')])
OrderedDict([('id', '9496328'), ('name', 'mahsa'), ('age', '24'), ('Field of Study', 'Computer Engineering')])
OrderedDict([('id', '9896325'), ('name', 'maryam'), ('age', '23'), ('Field of Study', 'computer science')])
OrderedDict([('id', '9496352'), ('name', 'sepideh'), ('age', '25'), ('Field of Study', 'electrical engineering')])
OrderedDict([('id', '9896389'), ('name', 'sima'), ('age', '26'), ('Field of Study', 'computer science')])
OrderedDict([('id', '9896325'), ('name', 'mina'), ('age', '21'), ('Field of Study', 'electrical engineering')])
```

استخدام pandas.read_csv()

قراءة ملف CSV باستخدام دوال مكتبة pandas أمر سهل وبسيط للغاية. تُستخدم طريقة read_csv() الخاصة بمكتبة pandas لقراءة بيانات ملف CSV.

In [1]:

```
import pandas

# reading the CSV file
csvFile = pandas.read_csv('data.csv')

# displaying the contents of the CSV file
print(csvFile)
```

Out [1]:

```
        id     name  age          Field of Study
0  9396321     sara   26  electrical engineering
1  9496328    mahsa   24    Computer Engineering
2  9896325   maryam   23        computer science
3  9496352  sepideh   25  electrical engineering
4  9896389     sima   26        computer science
5  9896325     mina   21  electrical engineering
```

في البرنامج أعلاه، تقوم طريقة read_csv() الخاصة بمكتبة pandas بقراءة ملف data.csv وتعيين بياناته إلى قائمة ثنائية الأبعاد.

XML (لغة توصيف النص القابلة للتوسعة[1])

تم تصميم XML بحيث يمكن قراءته بواسطة كل من البشر والآلات، لذلك يمكن استخدامه لتخزين البيانات ونقلها.في العالم الحقيقي، تحتوي أنظمة الكمبيوتر وقواعد البيانات على بيانات بتنسيقات غير متوافقة، لأن بيانات XML يتم تخزينهافي نص عادي، مما يوفر طريقة مستقلة عن البرامج والأجهزة لتخزين البيانات. هذا يجعل إنشاء البيانات التي يمكن مشاركتها بواسطة تطبيقات مختلفة أسهل بكثير. هنا مثال على صفحة XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
<food>
   <name>Belgian Waffles</name>
   <price>$5.95</price>
   <description>
  Two of our famous Belgian Waffles with plenty of real maple syrup
  </description>
   <calories>650</calories>
</food>
<food>
   <name>Strawberry Belgian Waffles</name>
   <price>$7.95</price>
   <description>
   Light Belgian waffles covered with strawberries and whipped cream
   </description>
   <calories>900</calories>
</food>
</breakfast_menu>
```

كيفية استيراد بيانات XML في بايتون

توفر لنا وحدة ElemenTree العديد من الأدوات لمعالجة ملفات XML. هذه الوحدة متاحة في مكتبة بايثون القياسية ، لذلك لا تحتاج إلى تثبيت أي وحدات خارجية لاستخدامها. توفر الوحدة النمطية ElementTree طرقًا لعرض مستند XML بالكامل على شكل شجرة. لقراءة ملف XML ، ندخل أولاً كلاس ElementTree في مكتبة XML. بعد ذلك ، انقل اسم ملف XML إلى دالة ElementTree.parse() لبدء التحليل. ثم نحصل على علامة جذر ملف XML باستخدام getroot(). ثم نعرض علامة جذر ملف XML. يتم استخدام root[0].attrib للحصول على خصائص العلامة الفرعية للجذر. بافتراض أن لدينا ملف XML على النحو التالي:

```
<model>
```

[1] eXtensible Markup Language

```
  <child name=”Acer” qty=”12”>Acer is a laptop</child>
  <unique>Add model number here</unique>
  <child name=”Onida” qty=”10”>Onida is an oven</child>
  <child name=”Acer” qty=”7”>Exclusive</child>
  <unique>Add price here</unique>
  <data>Add content here
    <family>Add company name here</family>
    <size>Add number of employees here</size>
  </data>
</model>
```

يوضح مقتطف الكود التالي كيفية قراءته باستخدام وحدة ElemenTree في بايثون:

```
In [1]: import xml.etree.ElementTree as ET

        # Pass the path of the xml document
        tree = ET.parse('data-text.xml')

        # get the parent tag
        root = tree.getroot()

        # print the root (parent) tag along with its memory
        location
        print(root)

        # print the attributes of the first tag
        print(root[0].attrib)

        # print the text contained within first subtag of the 5th
        tag from the parent
        print(root[5][0].text)
Out [1]:
        <Element 'model' at 0x0000028C44F2F548>
        {'name': 'Acer', 'qty': '12'}
        Add company name here
```

JSON (ترميز الكائنات باستعمال جافا سكريبت[1])

JSON هو تنسيق تبادل بيانات خفيف وشائع ليس من السهل على البشر قراءته وكتابته فحسب ، بل يسهل أيضًا على الآلات تحليله وإنشائه. إنه أيضًا أحد أكثر تنسيقات البيانات شيوعًا التي تستخدمها مواقع الويب عند نقل البيانات إلى JavaScript على الشاشة.

تم بناء JSON على هيكلين:

- **مجموعة من أزواج الاسم والقيمة.** في لغات مختلفة ، يُفهم هذا على أنه كائن أو سجل أو قاموس أو جدول تجزئة أو قائمة مفاتيح أو مصفوفة ارتباط.
- **قائمة قيم منظمة.** في معظم اللغات ، يُفهم هذا على أنه مصفوفة أو متجه أو قائمة أو تسلسل.

عندما يتم تبادل المعلومات بين المتصفح والخادم[2] ، يتم إرسال البيانات كنص فقط. JSON هو نص ، ويمكننا تحويل أي كائن JavaScript إلى JSON وإرسال JSON إلى الخادم. يمكننا أيضًا تحويل أي JSON تم استلامه من الخادم إلى كائنات JavaScript. بهذه الطريقة يمكننا العمل مع البيانات ككائنات جافا سكريبت بدون تحليل وترجمة معقدة. لنلقِ نظرة على بعض الأمثلة حول كيفية إرسال واستقبال البيانات باستخدام JSON

1. **ارسال البيانات:** إذا كانت البيانات مخزنة في كائن JavaScript ، فيمكننا تحويلها إلى JSON وإرسالها إلى الخادم. وفيما يلي مثال على ذلك:

```
<!DOCTYPE html>
  <html>
  <body>
  <p id="demo"></p>
  <script>
    var obj = {"name":"Milad", "age":29, "state": "Tehran"};
    var obj_JSON = JSON.stringify(obj);
    window.location = "json_Demo.php?x=" + obj_JSON;
  </script>
  </body>
  </html>
```

2. **استلام البيانات:** إذا كانت البيانات المستلمة بتنسيق JSON ، فيمكننا تحويلها إلى كائن JavaScript. على سبيل المثال:

```
<!DOCTYPE html>
   <html>
   <body>
   <p id="demo"></p>
   <script>
     var obj_JSON = "{"name":"Milad", "age":29, "state":  "Tehran"}";
```

---

[1] JavaScript Object Notation

[2] server

```
    var obj = JSON.parse(obj_JSON);
    document.getElementById("demo").innerHTML=obj.name;
  </script>
  </body>
  </html>
```

كيفية إدخال بيانات JSON في بايثون

تحميل كائن JSONفي بايثون سهل للغاية. لدى بايثون حزمة داخلية تسمى JSON يمكن استخدامها للعمل مع بيانات JSON. توفر لنا وحدة JSON هذه العديد من الطرق، من بينها طريقة loads() تساعدنا على قراءة ملف JSON. بافتراض أن لدينا ملف JSON على النحو التالي:

```
[
	{
		"Name": "Debian",
		"Version": "9",
		"Install": "apt",
		"Owner": "SPI",
		"Kernel": "4.9"
	},
	{
		"Name": "Ubuntu",
		"Version": "17.10",
		"Install": "apt",
		"Owner": "Canonical",
		"Kernel": "4.13"
	},
	{
		"Name": "Fedora",
		"Version": "26",
		"Install": "dnf",
		"Owner": "Red Hat",
		"Kernel": "4.13"
	},
	{
		"Name": "CentOS",
		"Version": "7",
		"Install": "yum",
		"Owner": "Red Hat",
		"Kernel": "3.10"
	},
	{
		"Name": "OpenSUSE",
		"Version": "42.3",
		"Install": "zypper",
		"Owner": "Novell",
		"Kernel": "4.4"
	},
	{
```

```
                "Name": "Arch Linux",
                "Version": "Rolling Release",
                "Install": "pacman",
                "Owner": "SPI",
                "Kernel": "4.13"
        },
        {
                "Name": "Gentoo",
                "Version": "Rolling Release",
                "Install": "emerge",
                "Owner": "Gentoo Foundation",
                "Kernel": "4.12"
        }
]
```


يوضح مقتطف الكود التالي كيفية قراءته باستخدام وحدة JSONفي بايثون:

```
In [1]: import json
        # Opening JSON file
        json_data = open('data.json').read()
        # returns JSON object as
        # a dictionary
        data = json.loads(json_data)
        # Iterating through the json
        # list
        for item in data:
            print (item)
Out [1]:
        {'Name': 'Debian', 'Version': '9', 'Install': 'apt', 'Owner': 'SPI', 'Kernel': '4.9'}
        {'Name': 'Ubuntu', 'Version': '17.10', 'Install': 'apt', 'Owner': 'Canonical', 'Kernel': '4.13'}
        {'Name': 'Fedora', 'Version': '26', 'Install': 'dnf', 'Owner': 'Red Hat', 'Kernel': '4.13'}
        {'Name': 'CentOS', 'Version': '7', 'Install': 'yum', 'Owner': 'Red Hat', 'Kernel': '3.10'}
        {'Name': 'OpenSUSE', 'Version': '42.3', 'Install': 'zypper', 'Owner': 'Novell', 'Kernel': '4.4'}
        {'Name': 'Arch Linux', 'Version': 'Rolling Release', 'Install': 'pacman', 'Owner': 'SPI', 'Kernel': '4.13'}
        {'Name': 'Gentoo', 'Version': 'Rolling Release', 'Install': 'emerge', 'Owner': 'Gentoo Foundation', 'Kernel': '4.12'}
```

المعالجة المسبقة للبيانات وتحضيرها

من غير المعقول توقع اكتمال البيانات. قد تكون هناك مشاكل بسبب خطأ بشري أو عيوب في عملية جمع البيانات. قد لا تكون بعض القيم موجودة، وفي حالات أخرى، قد توجد كائنات مزيفة أو مكررة. على سبيل المثال، قد تكون هناك حالتان مختلفتان لشخص عاش مؤخرًا في عنوانين مختلفين. حتى إذا كانت جميع البيانات متوفرة وتبدو جيدة، فقد يكون هناك تناقضات، على سبيل المثال، يبلغ طول الشخص مترين، لكنه يزن 2 كجم فقط.

بالإضافة إلى ذلك، من النادر جدًا أن تكون مجموعات البيانات متاحة كما هو مطلوب بواسطة خوارزميات علم البيانات. تتطلب معظم خوارزميات علم البيانات هياكل بيانات في شكل جداول بها سجلات في صفوف وخصائص في أعمدة. إذا كانت البيانات بتنسيق آخر، فيجب تعيين البيانات بحيث تصبح البيانات هي البُنية المطلوبة. لذلك، يجب تنظيف البيانات وتحويلها.

تنظيف البيانات

تنظيف البيانات عملية إعداد البيانات للتحليل عن طريق إزالة أو تعديل البيانات غير الصحيحة أو غير الكاملة أو غير الملائمة أو المكررة أو غير المناسبة. عادة ما تكون هذه البيانات غير ضرورية أو مفيدة في تحليل البيانات، لأنها قد تعطل العملية أو تقدم نتائج غير دقيقة. هناك عدة طرق لتنظيف البيانات، اعتمادًا على كيفية تخزين المعلومات والاستجابات. تنظيف البيانات لا تتعلق فقط بمسح المعلومات لإفساح المجال لبيانات جديدة، إنها تتعلق بإيجاد طريقة لتعظيم دقة مجموعة البيانات. يعتبر تنظيف البيانات عنصرًا أساسيًا في أساسيات علم البيانات لأنه يلعب دورًا مهمًا في عملية التحليل واكتشاف الإجابات الموثوقة. البيانات غير الصحيحة أو غير المتسقة تؤدي إلى نتائج غير صحيحة. لذلك، فإن طريقة تنظيف البيانات وفهمها لها تأثير كبير على جودة النتائج.

على سبيل المثال، قد ترغب الحكومة في تحليل إحصاءات التعداد لتحديد المجالات التي تحتاج إلى مزيد من الإنفاق والاستثمار في البُنية التحتية والخدمات. في هذه الحالة، سيكون الوصول إلى البيانات الصحيحة مهمًا لمنع القرارات المالية الخاطئة. أو في عالم الأعمال، قد تكون البيانات غير الصحيحة مكلفة. تستخدم العديد من الشركات مجموعات بيانات معلومات العملاء التي تسجل معلومات مثل معلومات الاتصال والعناوين. على سبيل المثال، إذا كانت العناوين غير متسقة، فستتحمل الشركة تكلفة إعادة إرسال الخطاب أو حتى فقدان العملاء.

يمكن لخوارزمية بسيطة التغلب على خوارزمية معقدة لمجرد أنها تحصل على بيانات كافية وعالية الجودة.

القيم المفقودة [1]

في بعض الأحيان قد تكون البيانات بالتنسيق الصحيح، ولكن بعض القيم مفقودة. ضعفي اعتبارك جدولًا يحتوي على معلومات العميل التي لا تتضمن بعض أرقام هواتف المنزل. قد يكون هذا بسبب أن بعض الأشخاص ليس لديهم هاتف منزلي وبدلاً من ذلك يستخدمون هواتفهم المحمولة كهاتف رئيسي.في حالات أخرى، قد تفقد البيانات بسبب مشاكلفي عملية جمع البيانات. بالإضافة إلى ذلك، قد لا تعتبر الشمولية مهمةفي وقت التجميع. على سبيل المثال، عندما بدأنافي جمع معلومات العميل، كان ذلك مقصورًا على مدينة أو منطقة معينة، لذلك لم يكن من الضروري جمع رمز المنطقة لرقم هاتف. الآن، قد نواجه مشاكل عندما نقرر التطوير خارج تلك المدينة أو المنطقة. إذن ماذا نفعل عندما نواجه بيانات مفقودة؟ لا توجد أفضل إجابة واحدة. علينا أن نجد الإستراتيجية الصحيحة بناءً على الظروف. وجود قيم مفقودةفي بياناتك ليس بالضرورة انتكاسة. ومع ذلك، فهذه فرصة لأداء هندسة الميزات المناسبة لتوجيه النموذج من أجل تفسير البيانات المفقودة بشكل صحيح. هناك عدة طرق مختلفة للتعامل مع هذه المشكلة، لكن لكل طريقة مزايا وعيوب. تتمثل الخطوة الأولىفي إدارة القيم المفقودةفي فهم سبب عدم وجود قيم. يمكن أن يؤدي تتبع مصدر البيانات إلى تحديد المشكلات النظامية أثناء إدخال البيانات أو أخطاء التحويل. غالبًا ما تخبرك معرفة مصدر القيمة المفقودة بالطريقة التي يجب استخدامها. يمكن استبدال القيمة المفقودة بمجموعة واسعة من البيانات الاصطناعية لإدارة المشكلة مع تأثير ضئيلفي المراحل اللاحقة من عملية علم البيانات. يتم سرد العديد من الاستراتيجيات المختلفة لإدارة البيانات المفقودة أدناه:

- **حذف الكائنات أو خصائص البيانات:** إستراتيجية بسيطة وفعالة لحذف الكائنات ذات القيم المفقودة. ومع ذلك، حتى كائن البيانات يحتوي على بعض المعلومات، وإذا كان عددًا كبيرًا من الكائنات يحتوي على قيم مفقودة، فقد يكون التحليل الموثوق به صعبًا أو مستحيلًا. ومع ذلك، إذا كان عدد قليل من الكائناتفي مجموعة البيانات تحتوي على قيم مفقودة، فقد يكون حذفها مفيدًا. تتمثل الإستراتيجية ذات الصلةفي حذف السمات التي تحتوي على قيم مفقودة. ومع ذلك، يجب أن يتم ذلك بحذر، حيث قد تكون الميزات المحذوفة ميزة مهمةفي التحليل.
- **تقدير القيم المفقودة:**في بعض الأحيان يمكن تقدير البيانات المفقودة بشكل موثوق. على سبيل المثال، ضعفي اعتبارك سلسلة زمنية تتغير منطقيًا ولكن بها بعض القيم المبعثرة المفقودة.في مثل هذه الحالات، يمكن تقدير القيم المفقودة (استكمال[2]) باستخدام القيم المتبقية. كمثال آخر، ضعفي اعتبارك مجموعة البيانات التي تحتوي على العديد من نقاط

[1] Missing Values

[2] interpolated

البيانات المتشابهة.في هذه الحالة، غالبًا ما تُستخدم قيم النقطة القريبة ذات القيمة المفقودة لتقدير القيمة المفقودة. إذا كانت السمة متصلة، فسيتم استخدام متوسط قيمة سمة الجيران الأقرب. إذا كانت الصفة منفصلة، فيمكن اعتبار القيمة الأكثر شيوعًا للسمة.

- **تجاهل القيم المفقودة أثناء التحليل:** هناك طريقة أخرى للتعامل مع البيانات المفقودة وهي تجاهل هذه القيم. على سبيل المثال، افترض أن الكائنات مجمعة ويجب حساب التشابه بين أزواج كائنات البيانات. إذا كان أحد الكائنات أو كلاهما يحتوي على قيم مفقودة لبعض الخصائص، فلا يمكن حساب التشابه إلا باستخدام الخصائص التي لا تحتوي على قيم مفقودة.

البيانات البعيدة (المتطرفة)[1]

في التعلم الآلي، لا تقل جودة البيانات أهمية عن جودة النموذج أو التصنيف التنبئي. ومع ذلك، في بعض الأحيانفي مجموعة البيانات، توجد بيانات مسجلة تختلف اختلافًا كبيرًا عن الحالات الأخرى، وتميز نفسهافي ميزة واحدة أو أكثر. هذه البيانات، المعروفة باسم البيانات المتطرفة، يمكن (وربما ستؤدي) إلى حدوث حالات شاذةفي النتائج التي تم الحصول عليها من خلال الخوارزميات والأنظمة التحليلية.في النماذج الخاضعة للاشراف، يمكن أن تخدع البيانات المتطرفة عملية التدريب ، مما قد يؤدي إلى فترات تدريب أطول أو يؤدي إلى نماذج أقل دقة.

إن قابلية تفسير نموذج بقيم متطرفة والتعرف على البيانات المتطرفة لهما جانبان مهمان للغاية في تحليل البيانات. أولاً ، على الرغم من البيانات المتطرفة ، قد تكون النتيجة بأكملها عبارة عن تحليل تحيز سلبي. ثانيًا ، قد يكون سلوك البيانات المتطرفة هو بالضبط ما نبحث عنه. في الواقع، يمكن أن تكون البيانات المتطرفة أحيانًا مؤشرات مفيدة. على سبيل المثال ، في بعض تطبيقات تحليلات البيانات ، مثل اكتشاف الاحتيال في بطاقة الائتمان ، يعد تحليل البيانات عن بُعد أمرًا مهمًا ، حيث قد يكون الاستثناء وليس القاعدة موضع اهتمام المحلل.

لا يمكن اعتبار البيانات المتطرفة تشوه أو خطأ. ومع ذلك، يشتبه في أنها لم يتم إنشاؤها بطريقة مماثلة للبيانات الأخرى (الكائنات).

أسباب توليد البيانات المتطرفة

فيما يلي بعض الأسباب الشائعة لوجود نقاط متطرفة في مجموعة بيانات معينة:

- **خطأ القياس (خطأ الادوات):** يحدث عندما تكون أداة القياس المستخدمة معيبة.
- **أخطاء إدخال البيانات (الأخطاء البشرية):** يمكن أن تسبب الأخطاء البشرية ، مثل الأخطاء التي تحدث أثناء جمع البيانات أو تسجيلها أو إدخالها ، مسافات طويلة في البيانات.

[1] Outliers

- **خطأ تجريبي:** تحدث هذه الأخطاء أثناء استخراج البيانات أو أثناء الاختبار.
- **خطأ معالجة البيانات:** يحدث عند معالجة مجموعات البيانات أو استخراجها.
- **خطأ في أخذ العينات:** يحدث هذا الخطأ عندما يقوم شخص ما باستخراج البيانات أو مزجها من مصدر خاطئ أو من مصادر مختلفة.
- **مقصودة:** هذه حالات خارجية مزيفة مصممة لاختبار طرق التشخيص.
- **طبيعية:** في عملية إنتاج البيانات وجمعها ومعالجتها وتحليلها ، يمكن أن تأتي النقاط المتطرفة من مصادر مختلفة وتكون مخفية في أبعاد مختلفة. البيانات التي ليست نتاج خطأ يطلق عليها متطرفة طبيعية.

تأثير البيانات المتطرفة على التحليل

البيانات المتطرفة لها تأثير كبير على نتيجة تحليل البيانات. فيما يلي بعض التأثيرات الأكثر شيوعًا:

- **قد يكون لها تأثير كبير على المتوسط والانحراف المعياري.**
- **إذا لم يتم توزيع تشتت النقاط المتطرفة بشكل عشوائي ، فيمكنها تقليل الحالة الطبيعية[1].**
- **يمكن أن تسبب التحيز (Bias) أو تؤثر على التقديرات.**
- **يمكن أن تؤثر على الافتراض الأساسي للتوقع (الانحدار) والنماذج الإحصائية الأخرى.**

انواع البيانات المتطرفة

في علم البيانات والإحصاء ، تنقسم البيانات المتطرفة عمومًا إلى ثلاث فئات رئيسية:

1. **البيانات العالمية البعيدة [2] (النقاط الشاذة):** هذه هي أبسط أشكال البيانات المتطرفة. إذا انحرفت نقطة بيانات واحدة بشكل حاد عن باقي نقاط البيانات في مجموعة بيانات معينة ، فإنها تُعرف باسم الناتج العالمي. على سبيل المثال ، في نظام الكشف عن التطفل، إذا تم توزيع عدد كبير من الحزم في فترة زمنية قصيرة جداً ، فيمكن اعتبار ذلك ناتجًا عالميًا ويمكن القول إن النظام المعين قد تم اختراقه.
2. **البيانات البعيدة موجه المحتوى [3] (الشرطية):** إذا كان كائن البيانات في مجموعة بيانات معينة ينحرف عن أجزاء أخرى من البيانات بناءً على محتوى (سياق) أو شروط محددة فقط. قد تكون نقطة البيانات بعيدة جداً بسبب موقف معين وتتصرف بشكل طبيعي في مواقف أخرى. لذلك ، يجب تحديد المحتوى كجزء من بيان المشكلة من أجل تحديد

[1] normality

[2] Global outliers

[3] Contextual outliers

نقاط المحتوى متطرفة. يتم تحديد خصائص نقطة البيانات بناءً على خصائص المحتوى والسلوك. على سبيل المثال ، قد تعمل درجة حرارة 40 درجة مئوية كدرجة حرارة متطرفة في محتوى "الشتاء" ، ولكنها تتصرف كنقطة بيانات عادية في محتوى "الصيف".

3. **البيانات الجماعية المتطرفة**[1]**:** إذا كانت بعض نقاط البيانات في مجموعة بيانات معينة تنحرف بشكل كبير عن بقية مجموعة البيانات ، فقد يشار إليها على أنها نقاط جماعية متطرفة. وتجدر الإشارة إلى أن قيم نقاط البيانات هذه ليست غير عادية بشكل فردي في المحتوى أو بشكل عام. لتمييز هذا عن المواقع البعيدة ، قد نحتاج إلى بيانات سابقة عن العلاقة بين كائنات البيانات تلك التي تشير إلى السلوكيات البعيدة.

كشف النقاط المتطرفة

هناك عدة طرق للعثور على المواقع المتطرفة. تستخدم كل هذه الطرق أسلوبًا للعثور على قيم غير معتادة مقارنة بمجموعات البيانات الأخرى. هنا قمنا بإدراج عدد قليل من هذه التقنيات:

- **الترتيب.** الترتيب هو أبسط تقنية لتحليل البيانات المتطرفة. قم بتحميل مجموعة البيانات الخاصة بك إلى أي نوع من أدوات معالجة البيانات ، مثل جدول بيانات (أو جدول) ، وقم بفرز القيم حسب الحجم. بعد ذلك ، تحقق من نطاق قيم نقاط البيانات المختلفة. إذا كانت كل نقطة بيانات أعلى أو أقل بشكل ملحوظ من النقاط الأخرى في مجموعة البيانات ، فيمكن اعتبارها عنصرًا بعيدًا. طريقة فرز البيانات على مجموعة بيانات صغيرة فعالة للغاية.
- **باستخدام الرسوم البيانية.** طريقة أخرى لتحليل البيانات البعيدة هي الرسوم البيانية. ارسم جميع نقاط البيانات على الرسم البياني واعرف النقاط الأبعد عن النقاط الأخرى. باستخدام طريقة الرسم التخطيطي مقارنة بطريقة الترتيب ، يمكننا تصور المزيد من نقاط البيانات التي تسهل رؤية النقاط المتطرفة. يمكننا تحديد النقاط المتطرفة باستخدام المخططات الصندوقية[2] ومخططات المدرج التكراري [3] ومخططات التشتت [4].
- **استخدام درجة z.** درجة z (الدرجة المعيارية) هي مقياس لعلاقة النقطة بمتوسط جميع النقاط في مجموعة البيانات. عند تحقيق الهدف، تتلقى القيم عددًا موجبًا أو سالبًا. من خلال حساب درجة z[5] لكل نقطة بيانات ، من السهل معرفة نقاط البيانات التي تتحرك بعيداً عن المتوسط في المتوسط. تفترض هذه الطريقة أن المتغير له توزيع غاوسي.

---

[1] Collective outliers

[2] boxplot

[3] histogram

[4] scatter plot

[5] $z = \frac{x-\mu}{\sigma}$

في هذه المعادلة ، x هي الدرجة الأولية ، و μ هي متوسط العينات و σ الانحراف المعياري للعينات.

البيانات المكررة[1]

غالبًا ما تحدث الملاحظات المكررة أثناء جمع البيانات. هناك فرص لإنشاء بيانات مكررة عند دمج مجموعات البيانات من مواقع متعددة أو جمع البيانات من خلال تصفح الويب أو تلقي البيانات من العملاء أو الفروع المتعددة. يعد حذف هذه البيانات المكررة أحد أكبر المشكلات التي يجب مراعاتهافي هذه العملية.

البيانات غير الضرورية[2]

البيانات غير ذات الصلة أو غير الضرورية هي تلك البيانات غير المطلوبة حقًا وليست مناسبةفي سياق المشكلة التي نحاول حلها. على سبيل المثال، إذا قمنا بتحليل بيانات الصحة العامة، فإن رقم الهاتف غير ذي صلة. كمثال آخر، إذاكنت تريد تحليل بيانات الجيل الجديد، لكن مجموعة البيانات الخاصة بك تتضمن الأجيال القديمة، فيجب عليك حذف تلك الملاحظات غير ذات الصلة. هذا يمكن أن يجعل التحليل أكثر كفاءة ويقلل من الالتباس حول هدفك الرئيسي.

يمكنك فقط حذف جزء من البيانات (سمة) غير ذي صلة. خلاف ذلك، اكتشف مصفوفة الارتباط بين متغيرات السمة، وحتى إذا لم تلاحظ أي ارتباط، يجب أن تسأل شخصًا متخصصًا في هذا المجال. ربما تكون الميزة التي تبدو غير ملائمة لك، وثيقة الصلة جداً من منطقة اخرى مثل المنظور السريري.

تحويل البيانات[3]

يجب تحويل البيانات إلى نظام قابل للقراءة ومتوافق. فيما يلي بعض العمليات الهامة التي يتم استخدامها لتحويل البيانات.

التجميع[4]

تجميع البيانات هو طريقة يتم فيها جمع البيانات الأولية واستخدامها بإيجاز للتحليل. على سبيل المثال، يمكن جمع البيانات الأولية خلال فترة زمنية لتقديم إحصائيات مثل المتوسط ، والحد الأدنى، والحد الأقصى، والمجموع. بعد تجميع البيانات وكتابتهاكتقرير، يمكنك تحليل البيانات المجمعة لاكتساب رؤى حول مصادر محددة. بمعنى آخر، يمكن أن يمكّن تجميع البيانات المحللين من الوصول إلى كميات كبيرة من البيانات وفحصهافي فترة زمنية معقولة. يمكن أن يمثل صف من البيانات المجمعة المئات أو الآلاف أو حتى أكثر من البيانات الدقيقة. ضعفي

---

[1] Duplicate Data

[2] Irrelevant data

[3] Data transformation

[4] Aggregation

اعتبارك مجموعة من البيانات التي تتضمن المعاملات التي تسجل المبيعات اليومية للمنتجات في مواقع مختلفةفي متجر لأيامفي السنة. تتمثل إحدى طرق تجميع المعاملاتفي مجموعة البيانات هذهفي استبدال جميع المعاملاتفي متجر بمعاملة واحدة. هذا يقلل من مئات أو آلاف المعاملات التي تحدث يوميًافي متجر معين ويقلل من عدد كائنات البيانات إلى عدد المتاجر.

تتضمن أمثلة البيانات المجمعة ما يلي:

- نسبة إقبال الناخبين حسب المقاطعة أو المدينة. لا يتم توفير سجلات الناخبين بشكل منفصل ، فقط العدد الإجمالي للأصوات لمرشح في دائرة انتخابية معينة.
- متوسط عمر العميل حسب المنتج. لم يتم تحديد كل عميل بشكل منفصل ، ولكن بالنسبة لكل منتج ، يتم تخزين متوسط عمر العميل.
- عدد العملاء حسب الدولة. بدلاً من مراجعة كل عميل ، يتم توفير عدد من العملاء من كل بلد.

التقطيع[1]

غالبًا ما نواجه البيانات التي يتم جمعها من العمليات المستمرة مثل درجة الحرارة والضوء المحيط وسعر سهم الشركة. لكنفي بعض الأحيان نحتاج إلى تقسيم هذه القيم المستمرة إلى أجزاء أكثر قابلية للتحكم (لأن بعض خوارزميات التعلم الآلي، وخاصة خوارزميات التصنيف، تتطلب البيانات لتكون سمات دفعة). يسمى تعيين البيانات من القيم المستمرة إلى القيم المنفصلة **بالتقطيع**. بتعبير أدق، يعد تحديد البيانات طريقة لتحويل قيم ميزات البيانات المستمرة إلى مجموعة محدودة من المسافات مع الحد الأدنى من فقدان البيانات. يمكننا فهم هذا المفهوم بمساعدة مثال. لنفترض أن لدينا سمة مثل العمر بالقيم الواردة أدناه:

العمر | 1,5,9,4,7,11,14,17,13,18,19,31,33,36,42,44,46,70,74,78,77

يوضح الجدول أدناه هذه البيانات بعد التقطيع:

السمة	**العمر**	**العمر**	**العمر**	**العمر**
	1,5,4,9,7	11,14,17,13,18,19	31,33,36,42,44,46	70,74,77,78
بعد التقطيع	الاطفال	الشباب	الكبار	المسنين

التنعيم[2]

يتم تنفيذ تنعيم البيانات باستخدام خوارزميات متخصصة لإزالة الضوضاء من مجموعة البيانات. تسمح هذه العملية بإبراز أنماط البيانات المهمة. يمكن أن يساعد تنعيم البياناتفي التنبؤ

[1] Discretization

[2] Smoothing

بالاتجاهات. على الرغم من أن تنعيم البيانات يمكن أن يساعدفي كشف الأنماطفي البيانات المعقدة، إلا أن تنعيم البيانات لا يوفر بالضرورة تفسيرًا للموضوع أو الأنماط التي تساعدفي التعرف عليه.في بعض الأحيان، قد يؤدي تنعيم البيانات إلى إزالة نقاط البيانات القابلة للاستخدام. إذا كانت مجموعة البيانات موسمية ولا تعكس بشكل كامل الواقع الناتج عن نقاط البيانات، فقد يؤدي ذلك إلى تنبؤات غير صحيحة.

التحجيم[1]

التعلم الآلي مثل صنع عصير مختلط. إذا أردنا الحصول على أفضل عصير، فعلينا أن نخلط جميع الفواكه ليس وفقًا لحجمها ولكن وفقًا لنسبها المناسبة. وبالمثل،في العديد من خوارزميات التعلم الآلي، لجلب جميع الميزاتفي موقف واحد، يتعين علينا التحجيم حتى لا نؤثر على عدد كبير من النماذج لمجرد حجمها الكبير. تعد **ميزات التحجيم**[2]في التعلم الآلي واحدة من أهم الخطوات في معالجة البيانات قبل إنشاء نموذج التعلم الآلي. يمكن أن يفرق التحجيم بين نموذج التعلم الآلي السيئ والنموذج الأفضل.

أكثر تقنيات تحجيم السمات شيوعًا هي التوحيد القياسي[3] والتسوية[4]. يتم استخدام التوحيد عندما نريد تقييد قيمنا بين رقمين، عادة بين [0،1] أو [1،1−]. بينما يحول التسوية البيانات إلى متوسط صفر وتباين 1.

سبب تحجيم البيانات؟

خوارزميات التعلم الآلي ترى الأرقام فقط. ومن ثم، إذا كان هناك اختلاف كبيرفي نطاق الأرقام، فإنهم يضعون الافتراض الأساسي بأن الأرقامفي النطاق الأعلى لها مزايا. وهكذا، يبدأ هذا العدد الكبيرفي لعب دور أكثر حسماً أثناء تدريب النموذج. بالإضافة إلى ذلك، تعمل خوارزميات التعلم الآلي على الأرقام ولا تعرف ما يمثله الرقم. يبلغ وزنه 10 جرامات وسعره 10 دولارات، وهو يمثل شيئين مختلفين تمامًا، وهو أمر واضح للبشر، ولكن بالنسبة للنموذج، كلاهما يعتبر ميزة. لنفترض أن لدينا خاصيتين للوزن والسعر أن قيم الوزن لا يجب أن تكون أعدادًا أكبر. ومن ثم، تفترض الخوارزمية أنه نظرًا لأن الوزن أكبر من السعر، فإن الوزن أهم من السعر. لهذا السبب، تلعب هذه الأرقام الأكثر أهمية دورًا أكثر حسماًفي تدريب النموذج. لذلك، فإن تحجيم السمات مطلوب لإحضار جميع السماتفي موقف واحد دون أي أهمية أولية.

---

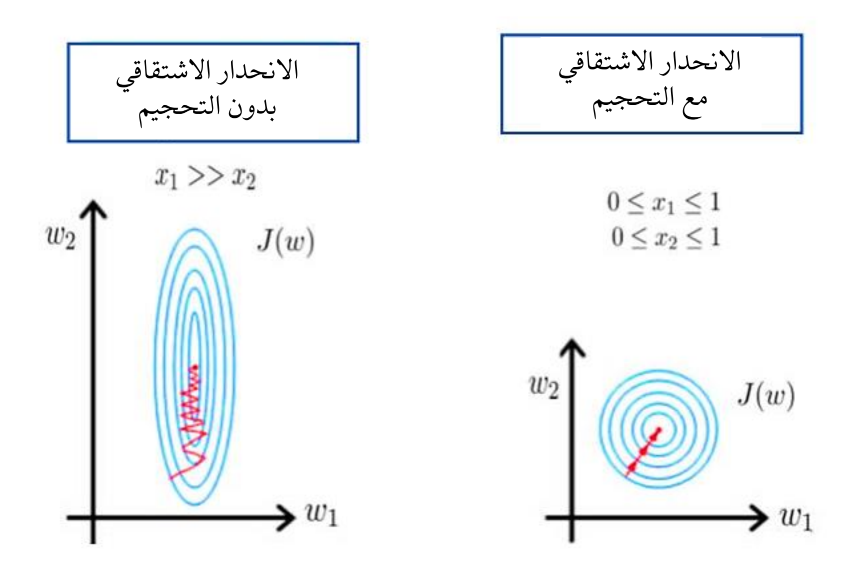
1 Scaling

2 Feature scaling

3 Standardization

4 normalization

سبب آخر لتطبيق تحجيم السمات هو أن عددًا صغيرًا من الخوارزميات، مثل الشبكة العصبية مع الانحدار الاشتقاقي، تتقارب بشكل أسرع مع تحجيم الميزة من دونها (الشكل 3ـ1).



الشكل 3ـ1 الانحدار الاشتقاقي مع/بدون التحجيم

التوحيد القياسي[1]

التوحيد القياسي للبيانات هو أسلوب مهم يتم إجراؤه كخطوة معالجة مسبقة قبل العديد من نماذج التعلم الآلي لتوحيد نطاق ميزات مجموعة بيانات الإدخال. يحدث التوحيد القياسي عندما تختلف خصائص مجموعة بيانات الإدخال اختلافًا كبيرًافي نطاقها. بعبارات أبسط، عندما يتم قياس البيانات بوحدات قياس مختلفة (على سبيل المثال، كيلوغرامات، أمتار، كيلومترات، إلخ). تسبب هذه الاختلافاتفي نطاق الميزات الأساسية مشاكلفي العديد من نماذج التعلم الآلي. على سبيل المثال، بالنسبة للنماذج التي تستند إلى حساب المسافة، إذا كان لإحدى الخصائص نطاق واسع من القيم، يتم ضبط المسافة بواسطة خاصية معينة. لنفترض أن لدينا مجموعة بيانات ثنائية الأبعاد بخاصيتين للطول بالأمتار والوزن بالكيلوجرام، والتي تتراوح بين [1 إلى 2] متر و [30 إلى 90] كجم، على التوالي. بغض النظر عن النموذج القائم على المسافة الذي تستخدمه بناءً على مجموعة البيانات هذه، فإن سمة الوزن ستسود على سمة الارتفاع وستكون لها حصة أكبر في حساب المسافة؛ فقط لأنه يحتوي على قيم أعلى مقارنة بالارتفاع. لذلك، لتجنب هذه المشكلة وحلها، من الضروري تحويل الميزات إلى مقاييس مماثلة باستخدام توحيد البيانات.

[1] Standardization

كيفية توحيد البيانات ؟

تعتبر الدرجة z ، والتي تسمى أيضًا الدرجة القياسية[1]، واحدة من أكثر الطرق شيوعًا لتوحيد البيانات، والتي يمكن إجراؤها عن طريق طرح المتوسط وتقسيمه على الانحراف المعياري لكل قيمة لكل سمة. معادلتها الرياضية على النحو التالي:

$$z = \frac{x - \mu}{\sigma}$$

في هذه المعادلة ، x هي الدرجة الأولية ، و μ هي متوسط العينة و σ هي الانحراف المعياري للعينة.

بمجرد اكتمال توحيد البيانات، سيكون لجميع السمات متوسط صفر، وانحراف معياري بواحد، وبالتالي، نفس المقياس.

متى يتم توحيد البيانات ؟

كما ذكرنا سابقًا، بالنسبة للنماذج القائمة على المسافة، يتم إجراء التوحيد لتجنب الميزات ذات النطاق الأوسع للتغلب على معيار المسافة. ومع ذلك، فإن سبب توحيد البيانات ليس هو نفسه بالنسبة لجميع نماذج التعلم الآلي ويختلف من نموذج إلى آخر. يميل بعض مطوري التعلم الآلي إلى تخصيص بياناتهم بشكل أعمى قبل أي نموذج للتعلم الآلي، دون محاولة معرفة سبب استخدامها. لذلك، قبل استخدام أي من نماذج وطرق التعلم الآلي، من الأفضل معرفة متى ولماذا يجب علينا استخدام التوحيد القياسي للبنات:

1. **قبل PCA:** في تحليل المكونات الرئيسية[2] (PCA) ، تكتسب الميزات ذات التباين العالي / النطاق الواسع وزنًا أكبر من تلك ذات التباين المنخفض ، ونتيجة لذلك ، فإنها تهيمن بشكل غير معقول على المكونات الرئيسية الأولى. يمكن أن يمنع التوحيد هذا من خلال توفير نفس الوزن لجميع الميزات.
2. **قبل التجميع:** نماذج التجميع[3] هي خوارزميات تعتمد على المسافة تستخدم معيار المسافة لقياس التشابه بين الملاحظات. لذلك ، سيكون للميزات عالية النطاق تأثير أكبر على التجميع. ومن ثم ، فإن التوحيد مطلوب قبل إنشاء نموذج التجميع.
3. **قبل KNN:** K_أقرب الجيران[4] هي خوارزمية تصنيف تعتمد على المسافة تصنف الملاحظات الجديدة بناءً على أوجه التشابه (على سبيل المثال، معايير المسافة) مع

[1] standard score

[2] Principal Component Analysis

[3] Clustering

[4] k-nearest neighbors

الملاحظات الموسومة من مجموعة التدريب. يسمح التوحيد لجميع المتغيرات بالمشاركة بالتساوي في قياس التشابه.

4. **قبل SVM**: تحاول خوارزمية آلة المتجهات الداعمة[1] تعظيم المسافة بين لوحة الفاصل ومتجهات الدعم. إذا كانت الخاصية تحتوي على قيم كبيرة جداً ، فإنها تهيمن على الخصائص الأخرى عند حساب المسافة. لذلك، يعطي التوحيد جميع الميزات نفس التأثير على معيار المسافة.

التسوية[2]

التسوية هو جزء من تقنيات المعالجة المسبقة وتنقية البيانات، وبشكل أكثر عمومية، نوع من تحجيم الميزات. الغرض الرئيسي من هذه التقنية هو جعل البيانات متسقة عبر جميع السجلات والحقول (دون تغيير نطاق القيم). يساعد هذافي إجراء اتصالات بين بيانات الإدخال، مما يساعد بدورهفي تنظيف البيانات وتحسين جودتها. يتم استخدام هذا النوع من التحجيم عندما يكون للبيانات نطاق متنوع (للخصائص نطاقات مختلفة) ولا تفترض الخوارزميات التي يتم تدريبها عليها مسبقًا توزيع البيانات (مثل الشبكات العصبية).

يعطي التسوية وزناً / أهمية متساوية لكل متغير بحيث لا يشوه متغير واحد أداء النموذج في اتجاه واحد ؛ فقط لأنهم أكثر عددا. إن أسلوب التحجيم الأكثر شيوعًا والأكثر استخدامًا هو **التحجيم مجدداً**[3] ، والمعروف أيضًا باسم تسوية الحد الاقل_الحد الاكثر، والذي يتم حسابه على النحو التالي:

$$\acute{x} = \frac{x - min(x)}{max(x) - min(x)}$$

التوحيد او التسوية؟

تعد التسوية مفيدة عندما تعلم أن توزيع البيانات الخاص بك لا يتبع التوزيع الغاوسي (منحنى الجرس). يمكن أن يكون هذا مفيداًفي الخوارزميات التي لا تفترض أي توزيع للبيانات، مثل KNN أو الشبكات العصبية. من ناحية أخرى، يمكن أن يكون التوحيد مفيداًفي الحالات التي تتبع فيها البيانات توزيعًا غاوسيًا (يفترض التوحيد أن بياناتك لها توزيع غاوسي). ومع ذلك، هذا ليس صحيحًا بالضرورة، ولكن إذا كان توزيع الميزات الخاص بك هو غاوسي، فإن هذه التقنية تكون أكثر فعالية. أيضًا، على عكس التوحيد، ليس للتسوية حدود. لذلك، حتى إذا كان

---

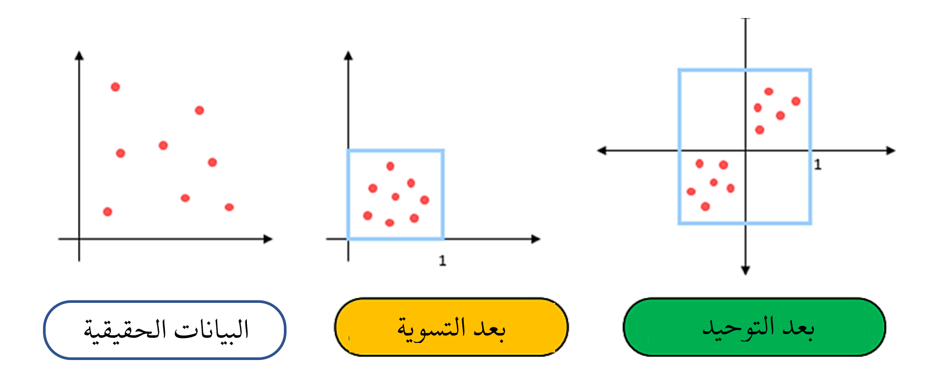
[1] Support Vector Machine

[2] normalization

[3] Rescaling

لديك الكثير من البيانات المتطرفةفي بياناتك، فلن تتأثر بالتسوية. ومع ذلك، يعتمد اختيار استخدام التوحيد أو التسوية على مشكلتك وخوارزمية التعلم الآلي الخاصة بك. لا توجد قاعدة صارمة وسريعة لإخبارك بموعد توحيد بياناتك. يمكنك دائمًا ملاءمة[1] نموذجك مع البيانات الخام والموحدة والمسواة ومقارنة الأداء للحصول على أفضل النتائج.

بينما يضع التوحيد القيم الأساسية ضمن نطاق معين، يضعها التسوية فى توزيع متوسطه صفر وانحرافه المعيارى واحد.



التمثيل المرئى للبيانات

التمثيل المرئي للبيانات هو عرض البيانات أو المعلومات في رسم بياني أو مخطط بياني أو أي تنسيق مرئي آخر ينقل اتصال البيانات بالصور. يعد التمثيل المرئي للبيانات أحد أهم الخطوات في علم البيانات والتعلم الآلي ، حيث يتيح رؤية الاتجاهات والأنماط بسهولة. مع ظهور البيانات الضخمة ، نحتاج إلى أن نكون قادرين على تفسير مجموعات كبيرة من البيانات. نحتاج إلى التمثيل المرئي للبيانات ، لأن الملخص المرئي للبيانات يجعل من السهل تحديد الأنماط والاتجاهات بدلاً من النظر إلى آلاف الصفوف في جدول بيانات. هذه هي الطريقة التي يعمل بها الدماغ البشري. نظرًا لأن الغرض من تحليل البيانات هو الحصول على نظرة ثاقبة ، يمكن الحصول على مزيد من المعلومات من البيانات عند توضيحها. حتى إذا تمكن محلل البيانات من استخلاص رؤى من البيانات دون توضيح ، فسيكون من الصعب نقل المفهوم دون تمثيل مرئي.

عندما يكتب عالم البيانات خوارزميات تحليل تنبؤي متقدمة، من المهم تصوير المخرجات لمراقبة النتائج وضمان أداء النماذج. هذا لأن التمثيل المرئي للخوارزميات المعقدة أسهل بشكل

[1] fitting

عام من المخرجات الرقمية. ربما يكون التمثيل المرئي للبيانات في بايثون أحد أكثر الميزات المفيدة المستخدمة في علم البيانات مع بايثون اليوم. في استمرار لهذا القسم ، بعد تقديم أنواع الرسوم البيانية التوضيحية للبيانات وأسباب اختيار كل منها ، سنقوم بتنفيذ كل من هذه المخططات باستخدام مكتبات بايثون القوية الموجودة في هذا المجال.

أهمية وفوائد التمثيل المرئى للبيانات

بغض النظر عن المهنة التي تختارها ، يمكن أن يساعد التمثيل المرئي للبيانات في عرض البيانات بأكثر الطرق فعالية. التمثيل المرئي يلتقط البيانات الأولية ، ويضعها في نماذج ، ويعرض البيانات للوصول إلى نتيجة. يمكن أن يساعد هذا الشركات في تحديد المجالات التي تحتاج إلى تحسين، وما هي العوامل التي تؤثر على رضا العملاء وعدم رضاهم ، وماذا تفعل بمنتجات معينة (إلى أين يجب أن يذهبوا ولمن يجب بيعها). تمنح البيانات المصورة أصحاب المصلحة وأصحاب الأعمال وصناع القرار توقعات أفضل للمبيعات والنمو في المستقبل.

لتمثيل البيانات مرئياً تأثير إيجابي على اتخاذ القرارات في المنظمات والشركات من خلال العرض المرئي التفاعلي للبيانات. يمكن للشركات الآن التعرف على الأنماط بشكل أسرع لأنها تستطيع تفسير البيانات بيانياً أو بصرياً. فيما يلي بعض الطرق المحددة التي يمكن أن تفيد المؤسسة من خلالها تصور البيانات:

- **الارتباط في العلاقات:** بدون التمثيل المرئي للبيانات ، يكون تحديد العلاقة بين علاقة المتغيرات المستقلة أمرًا صعبًا. من خلال فهم المتغيرات المستقلة ، يمكن اتخاذ قرارات عمل أفضل.
- **الاتجاهات بمرور الوقت:** يعد هذا أحد أكثر تطبيقات التمثيل المرئي للبيانات قيمةً. من المستحيل التنبؤ بدون الحصول على المعلومات اللازمة من الماضي والحاضر. تخبرنا الاتجاهات بمرور الوقت أين كنا ومن المحتمل إلى أين يمكننا الذهاب.
- **البحث عن الاسواق:** يأخذ التمثيل الصوري للبيانات معلومات من أسواق مختلفة لمنحك نظرة ثاقبة على الجمهور الذي يجب أن يكون تركيزك عليه وأي منهم يجب تجنبه. من خلال عرض هذه البيانات في رسوم بيانية مختلفة ، نحصل على صورة أوضح للفرص المتاحة في السوق.
- **استجابة السوق:** القدرة على الحصول على المعلومات بسرعة وسهولة باستخدام البيانات المعروضة بوضوح على سطح مكتب وظيفي ، مما يسمح للشركات بالتصرف بسرعة والاستجابة للنتائج ، والمساعدة في تجنب الأخطاء.

تشمل الفوائد الأخرى للتمثيل المرئي للبيانات ما يلي:

- القدرة على استيعاب المعلومات بسرعة وتحسين الرؤية واتخاذ القرارات بشكل أسرع.
- فهم المزيد حول الخطوات التالية التي يجب اتخاذها لتحسين المنظمة.
- سهولة توزيع المعلومات التي تزيد من فرصة مشاركة الأفكار مع جميع المعنيين.

ما هو الغرض من التمثيل المرئي للبيانات؟

الغرض من التمثيل المرئي للبيانات واضح تمامًا؛ إعطاء معنى للبيانات واستخدام المعلومات لصالح المنظمة. بالإضافة إلى ذلك، البيانات معقدة وتصبح أكثر قيمة كلما تم توضيحها. بدون التمثيل المرئي، من الصعب العثور بسرعة على اتصالات من البيانات وتحديد الأنماط لاكتساب البصيرة. يمكن لعلماء البيانات العثور على أنماط أو أخطاء دون توضيح. ومع ذلك، يعد نقل النتائج من البيانات وتحديد المعلومات الهامة منها أمرًا بالغ الأهمية. يعمل التمثيل المرئي للبيانات على تحسين تأثير الرسائل على جمهورك ويقدم نتائج تحليل البيانات بالطريقة الأكثر إقناعًا. يسمح لك التمثيل المرئي البيانات بفهم كميات كبيرة من البيانات بشكل أفضل في لمحة. يساعد هذافي فهم البيانات بشكل أفضل لقياس تأثيرها على الأعمال التجارية ونقل الأفكار بصريًا إلى الجمهور المحلي والأجنبي.

ما نوع الرسم البياني الذي يجب أن نستخدمه؟

قبل أن تبدأفي النظر إلى أنواع المخططات، عليك أن تسأل نفسك 5 أسئلة مهمة حول البيانات التي لديك. ستساعدك هذه الأسئلة على فهم بياناتك بشكل أفضل، وبالتالي اختيار نوع الرسم البياني المناسب لعرضها.

1. **ما هو الموضوع (التقرير) الذي تحاول بياناتك تقديمه؟**

أول شيء يجب أن تعرفه عن بياناتك هو ما الموضوع أو التقرير الذي تحاول تقديمه؟ لماذا تم جمع هذه البيانات وكيف؟ هل تم جمع بياناتك للعثور على الاتجاهات؟ لمقارنة الخيارات المختلفة؟ هل تُظهر التوزيع؟ أم أنها تستخدم لمراقبة العلاقة بين مجموعات القيم المختلفة؟ إن فهم مصدر بياناتك ومعرفة ما تحاول تقديمه سيسهل عليك اختيار نوع الرسم البياني.

2. **لمن ستقدم نتائجك؟**

بمجرد فهم المشكلة الكامنة وراء بياناتك،في الخطوة التالية، ستحتاج إلى معرفة من تقدم نتائجك إليه. إذاكنت تقوم بتحليل اتجاهات سوق الأسهم وتقديم نتائجك إلى بعض المتداولين، فيمكنك استخدام نوع مخطط مختلف عما كان عليه عندما استخدمت نتائجك للأشخاص الذين بدأوا للتو في سوق الأسهم. الهدف العام من استخدام التمثيل المرئي للبيانات هو جعل اتصال البيانات

أكثر كفاءة. لهذا السبب، تحتاج إلى معرفة جمهورك حتى تتمكن من اختيار أفضل مخطط لبياناتك.

3. ما هو حجم البيانات الخاصة بك؟

يؤثر حجم البيانات بشكل كبير على نوع الرسم البياني الذي تستخدمه. يجب عدم استخدام بعض الرسوم البيانية لمجموعات البيانات الكبيرة، بينما البعض الآخر مناسب لمجموعات البيانات الكبيرة. على سبيل المثال، تعمل المخططات الدائرية بشكل أفضل مع مجموعات البيانات الصغيرة. ومع ذلك، إذا كنت تستخدم مجموعة بيانات كبيرة، فمن المنطقي أكثر أن تستخدم مخطط التشتت. تحتاج إلى اختيار نوع الرسم البياني الذي يناسب حجم بياناتك ويظهرها بوضوح دون فوضى.

4. ما هو نوع البيانات الخاصة بك؟

هناك أنواع مختلفة من البيانات، اسمية، متسلسلة، مستمرة أو متقطعة. يمكنك استخدام نوع البيانات لحذف بعض أنواع المخططات. على سبيل المثال، إذا كانت لديك بيانات مستمرة، فقد لا يكون المخطط الشريطي هو الخيار الأفضل. قد تحتاج إلى استخدام مخطط خطي بدلاً من ذلك. وبالمثل، إذا كانت لديك بيانات متقطعة، فقد يكون من الجيد استخدام المخططات الشريطية أو المخططات الدائرية.

5. كيف ترتبط العناصر المختلفة لبياناتك ببعضها البعض ؟

أخيرًا، عليك أن تسأل نفسك كيف ترتبط العناصر المختلفة لبياناتك ببعضها البعض. هل يعتمد ترتيب بياناتك على بعض العوامل مثل الوقت والحجم والنوع؟ أو الارتباط بين المتغيرات المختلفة؟ هل بياناتك عبارة عن سلسلة زمنية؟ أم المزيد من التوزيع؟ تساعدك العلاقة بين القيم في مجموعة البيانات الخاصة بك على تحديد أفضل مخطط.

أنواع الرسوم البيانية للتمثيل المرئى للبيانات

الآن بعد أن عرفنا ما هو التمثيل المرئي للبيانات وكيف يعمل، دعونا نلقي نظرة على الأنواع المختلفة من الرسوم البيانية التي تقوم بالتمثيل المرئي للبيانات.

الرسم البيانى الخطى

يتم استخدام المخطط البياني الخطي لإظهار تغيير البيانات خلال فترة زمنية متصلة أو خلال فترة زمنية. بمعنى آخر، يتم استخدام الرسوم البيانية الخطية بشكل فعال عندما نريد فهم الاتجاهات بمرور الوقت.

متى نستخدم الرسم البياني الخطي ؟

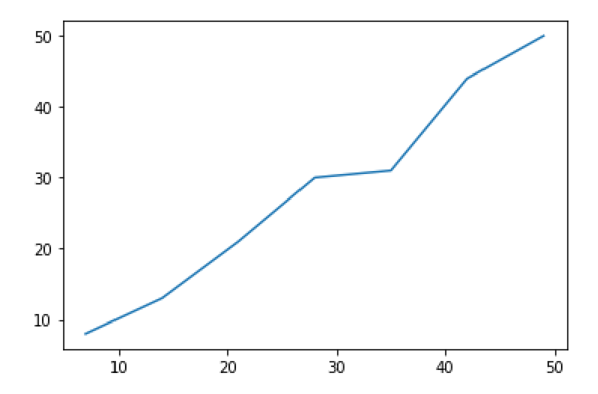
- إذا كان لديك مجموعة بيانات مستمرة تتغير بمرور الوقت.
- إذا كانت مجموعة البيانات الخاصة بك كبيرة جدًا بالنسبة إلى المخطط الشريطي.
- عندما تريد إظهار الاتجاهات لفئات مختلفة خلال نفس الفترة الزمنية وبالتالي إظهار المقارنات.
- رسم الاتجاهات بدلاً من القيم الدقيقة.

متى لا نستخدم الرسم البياني الخطي؟

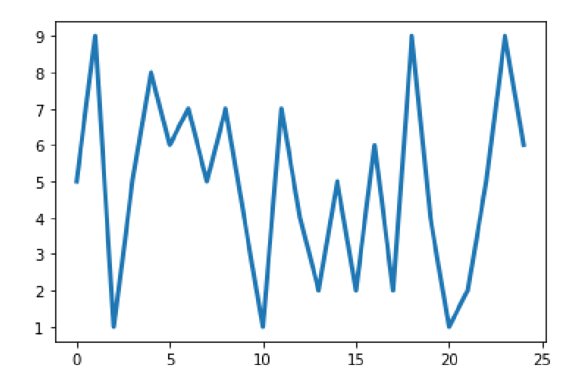
- يعمل الرسم البياني الخطي بشكل أفضل مع مجموعة بيانات أكبر، لذلك إذا كان لديك مجموعة بيانات صغيرة ، فاستخدم مخطط شريطي بدلاً من ذلك.

الرسم البياني الخطي باستخدام Matplotlib

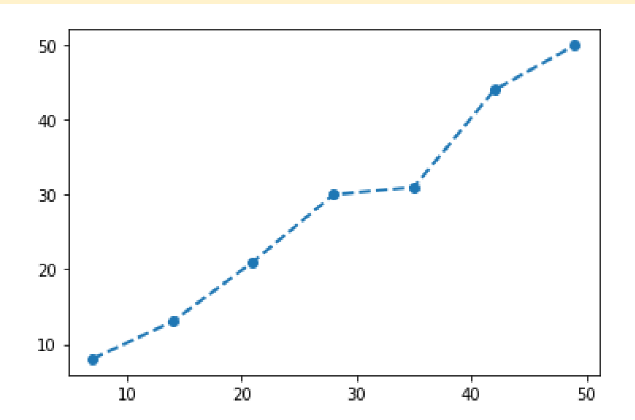
```
# Importing packages
import matplotlib.pyplot as plt
# Define x and y values
x = [7, 14, 21, 28, 35, 42, 49]
y = [8, 13, 21, 30, 31, 44, 50]
# Plot a simple line chart without any feature
plt.plot(x, y)
plt.show()
```




```
# Importing packages
import numpy as np
# Define x  value
x = np.random.randint(low=1, high=10, size=25)
plt.plot(x, linewidth=3)
plt.show()
```



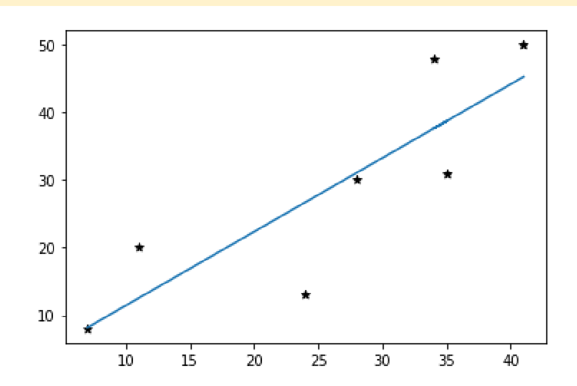

```
# Define x and y values
x = [7, 14, 21, 28, 35, 42, 49]
y = [8, 13, 21, 30, 31, 44, 50]
# Plot points on the line chart
plt.plot(x, y, 'o--', linewidth=2)
plt.show()
```



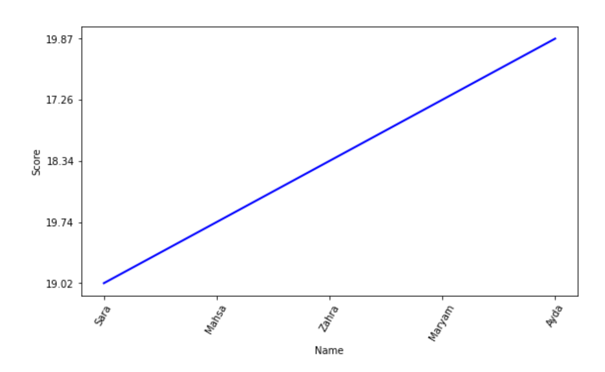

```
# Define x and y values
x = np.array([7, 11, 24, 28, 35, 34, 41])
y = np.array([8, 20, 13, 30, 31, 48, 50])

# Drawn a simple scatter plot for the data given
plt.scatter(x, y, marker='*', color='k')

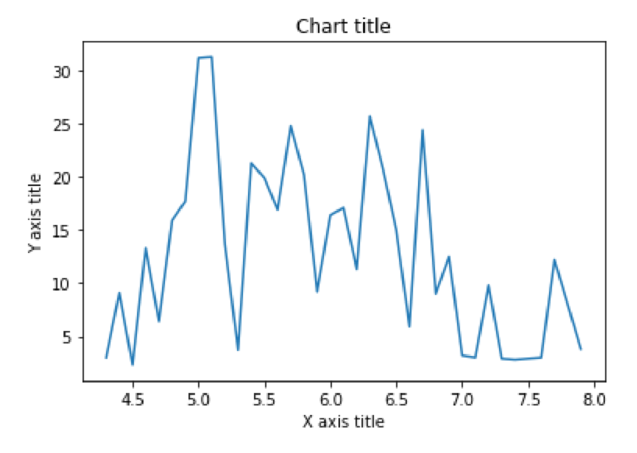
# Generating the parameters of the best fit line
m, c = np.polyfit(x, y, 1)
# Plotting the straight line by using the generated parameters
plt.plot(x, m*x+c)
plt.show()
```




```
import pandas as pd
# Let's create a Dataframe using lists
Name = ['Sara', 'Mahsa', 'Zahra', 'Maryam', 'Ayda']
Score = ['19.02', '19.74', '18.34', '17.26', '19.87']
# Now, create a pandas dataframe using above lists
df_ = pd.DataFrame(
    {'Name' : Name, 'Score' : Score})
# Plotting the data from the dataframe created using matplotlib
plt.figure(figsize=(9, 5))
plt.plot(df_['Name'], df_['Score'], '-b', linewidth=2)
plt.xticks(rotation=60)
plt.xlabel('Name')
plt.ylabel('Score')
plt.show()
```



```
#importing the required libraries
import matplotlib.pyplot as plt
import seaborn as sns
#Creating the dataset
df = sns.load_dataset("iris")
df=df.groupby('sepal_length')['sepal_width'].sum().to_frame().reset_index()
#Creating the line chart
plt.plot(df['sepal_length'], df['sepal_width'])
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
#Show the plot
plt.show()
```

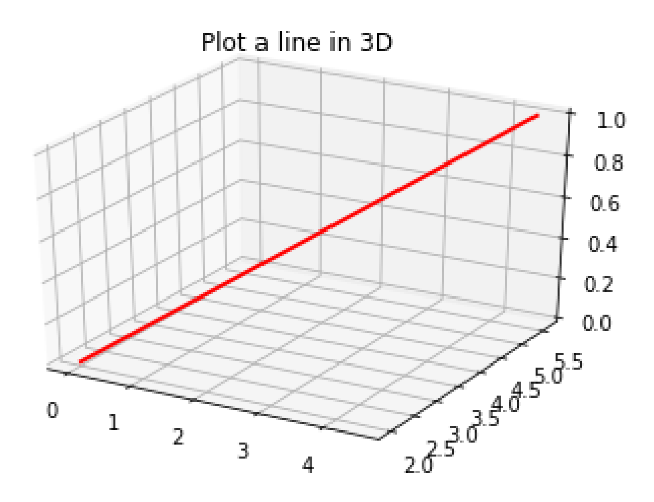



```
import numpy as np
from mpl_toolkits import mplot3d

# Setting 3 axes for the graph
plt.axes(projection='3d')

# Define the z, y, x data
z = np.linspace(0, 1, 100)
x = 4.5 * z
y = 0.8 * x + 2

# Plotting the line
plt.plot(x, y, z, 'r', linewidth=2)
plt.title('Plot a line in 3D')
plt.show()
```

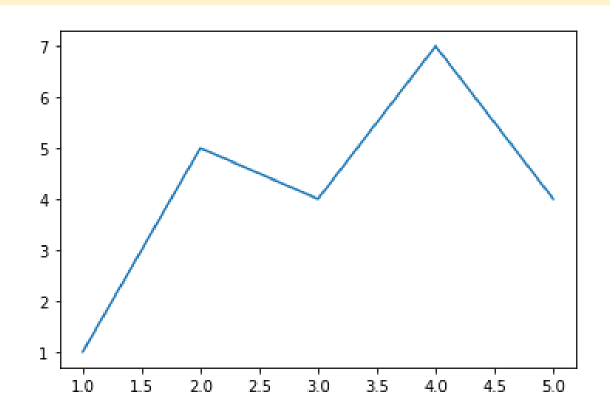


الرسم البياني الخطي باستخدام Seaborn

```
#importing the required libraries
import seaborn as sns

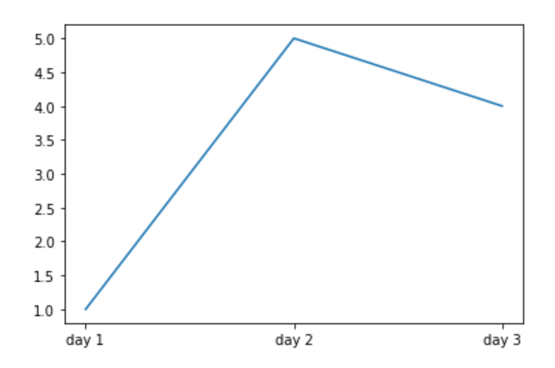
# Define the  x and y data
x = [1, 2, 3, 4, 5]
y = [1, 5, 4, 7, 4]

sns.lineplot(x, y)
plt.show()
```




```
# Define the  x and y data
x = ['day 1', 'day 2', 'day 3']
y = [1, 5, 4]

sns.lineplot(x, y)
plt.show()
```



الرسم البيانى الشريطى

المخططات الشريطية هي طريقة لعرض قيم البيانات المعروضة كأشرطة أفقية وتستخدم لإظهار اتجاه البيانات ومقارنة البيانات في مجموعات فرعية مختلفة معًا.

متى نستخدم الرسم البياني شريطي ؟

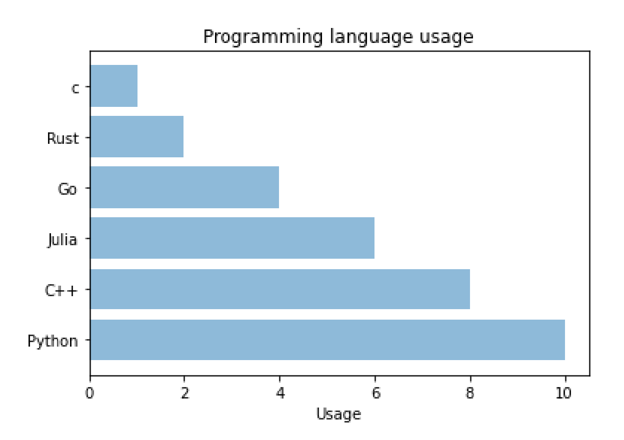
- عندما تحتاج إلى مقارنة عدة فئات مختلفة.
- عندما تحتاج إلى إظهار كيف تتغير البيانات الضخمة بمرور الوقت.
- إذا كنت تريد إظهار القيم الموجبة والسالبة في مجموعة البيانات.

متى لا نستخدم الرسم البياني الشريطي؟

- إذا كان لديك العديد من الفئات. يجب ألا يتجاوز المخطط الخاص بك 10 أشرطة.

الرسم البياني الخطي باستخدام Matplotlib

```
import numpy as np
import matplotlib.pyplot as plt
# Dataset generation
objects = ('Python', 'C++', 'Julia', 'Go', 'Rust', 'c')
y_pos = np.arange(len(objects))
performance = [10,8,6,4,2,1]
#  Bar plot
plt.barh(y_pos, performance, align='center', alpha=0.5)
plt.yticks(y_pos, objects)
plt.xlabel('Usage')
plt.title('Programming language usage')
plt.show()
```

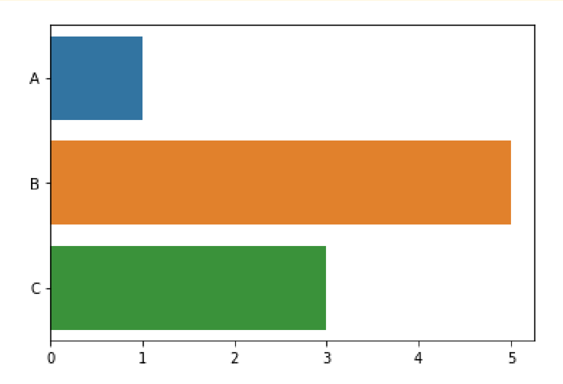


الرسم البياني الخطي باستخدام Seaborn

```
import matplotlib.pyplot as plt
import seaborn as sns

x = ['A', 'B', 'C']
y = [1, 5, 3]

sns.barplot(y, x)
plt.show()
```



المخطط العمودي

المخطط العمودي هو نوع من الرسومات البيانية الشريطية التي تستخدم أشرطة عمودية لإظهار المقارنات بين الفئات. من الأفضل استخدام المخططات الشريطية لعرض الموقف في وقت معين (على سبيل المثال ، عدد المنتجات المباعة على موقع ويب). الغرض الرئيسي منها هو لفت الانتباه إلى الأرقام بدلاً من الاتجاهات (الاتجاهات أكثر ملاءمة للرسوم البيانية الخطية).

متى نستخدم الرسم البياني العمودي؟

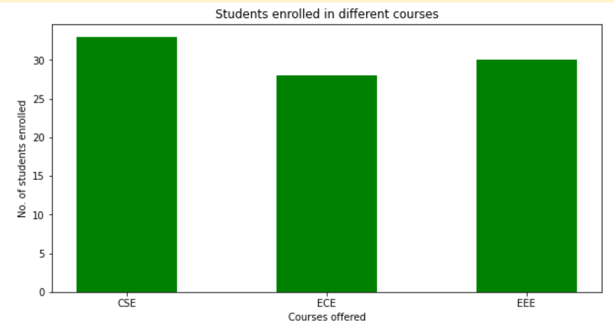
- اعرض قيمًا مختلفة عندما تحتاج إلى مقارنات جنبًا إلى جنب.
- عندما تريد التأكيد على الفرق بين القيم.
- عندما تريد إبراز جميع الأرقام بدلاً من الاتجاهات.

متى لا نستخدم الرسم البياني العمودي ؟

- مناسب لمجموعات البيانات الصغيرة والمتوسطة فقط.
- يجب ألا يكون عدد الأعمدة كبيرًا جداً.

الرسم البياني العمودي باستخدام Matplotlib

```
import numpy as np
import matplotlib.pyplot as plt
# Dataset generation
data_dict = {'CSE':33, 'ECE':28, 'EEE':30}
courses = list(data_dict.keys())
values = list(data_dict.values())
fig = plt.figure(figsize = (10, 5))
#  Bar plot
plt.bar(courses, values, color ='green',
        width = 0.5)
plt.xlabel("Courses offered")
plt.ylabel("No. of students enrolled")
plt.title("Students enrolled in different courses")
plt.show()
```



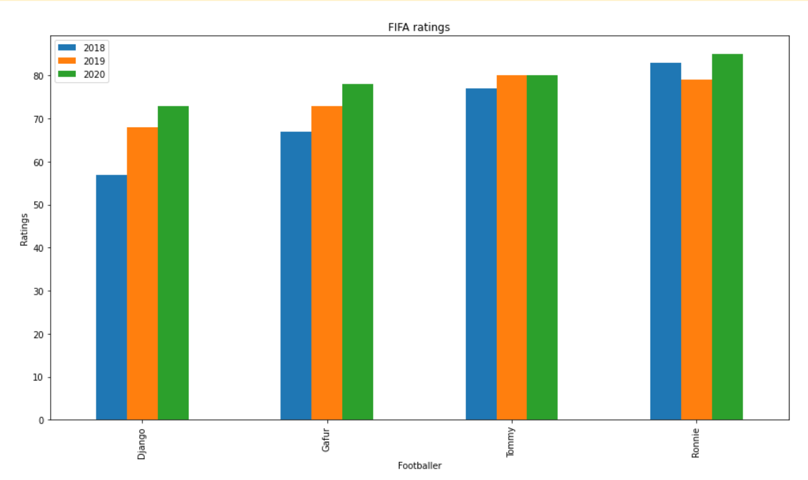
```
import pandas as pd
plotdata = pd.DataFrame({

    "2018":[57,67,77,83],

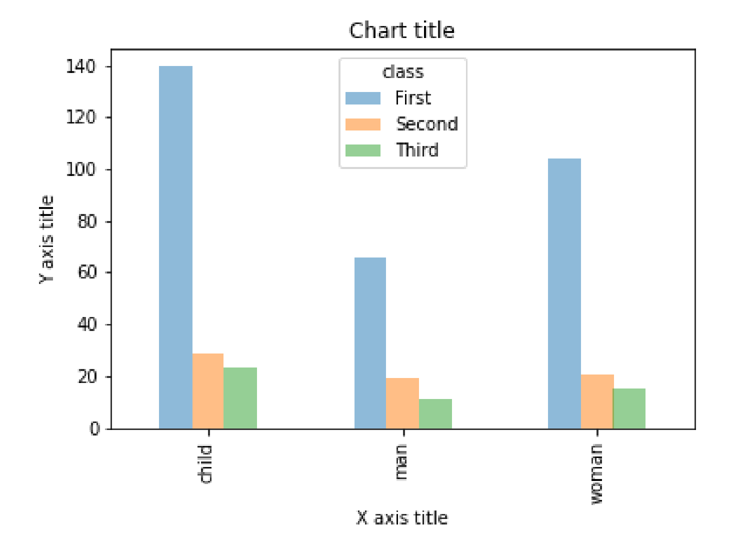
    "2019":[68,73,80,79],

    "2020":[73,78,80,85]},

    index=["Django", "Gafur", "Tommy", "Ronnie"])
plotdata.plot(kind="bar",figsize=(15, 8))
plt.title("FIFA ratings")
plt.xlabel("Footballer")
plt.ylabel("Ratings")
```

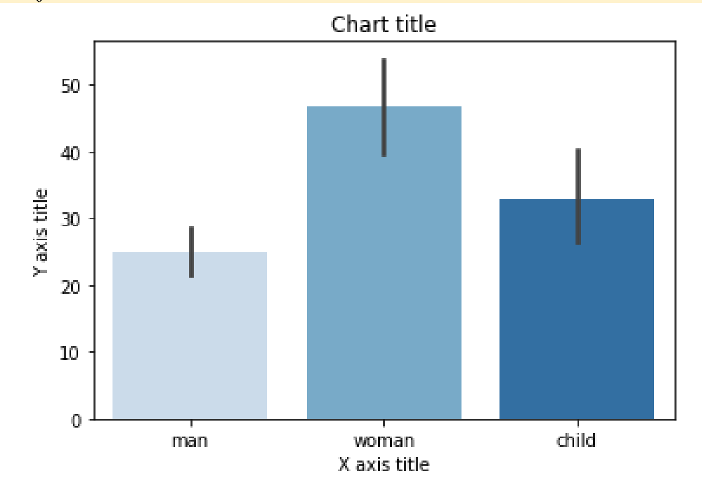



```
#Creating the dataset
df = sns.load_dataset('titanic')
df_pivot = pd.pivot_table(df,
values="fare",index="who",columns="class", aggfunc=np.mean)
#Creating a grouped bar chart
ax = df_pivot.plot(kind="bar",alpha=0.5)
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```

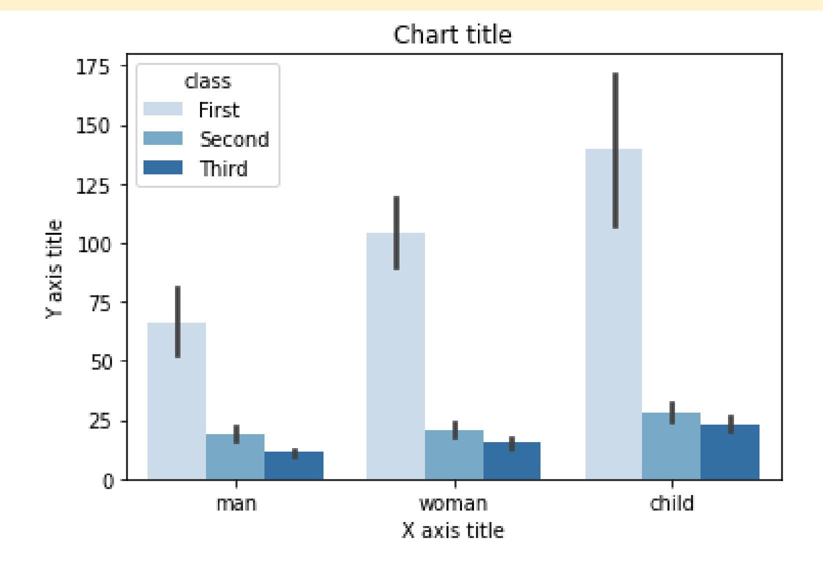


الرسم البياني العمودي باستخدام Seaborn

```
#Reading the dataset
titanic_dataset = sns.load_dataset('titanic')
#Creating column chart
sns.barplot(x = 'who',y = 'fare',data = titanic_dataset,palette = "B
lues")
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```



```
#Reading the dataset
titanic_dataset = sns.load_dataset('titanic')
#Creating the bar plot grouped across classes
sns.barplot(x = 'who',y = 'fare',hue = 'class',data = titanic_datase
t, palette = "Blues")
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```



الرسم البياني الدائري

تستخدم المخططات الدائرية على نطاق واسع في مختلف المجالات لإظهار نسب التصنيفات المختلفة ولمقارنة التصنيفات المختلفة بالقوس. تعد المخططات الدائرية خيارًا رائعًا لتمثيل النسب المئوية ، لأنها تمثل كل عنصر كجزء من الكل.

متى نستخدم الرسم البياني الدائري؟

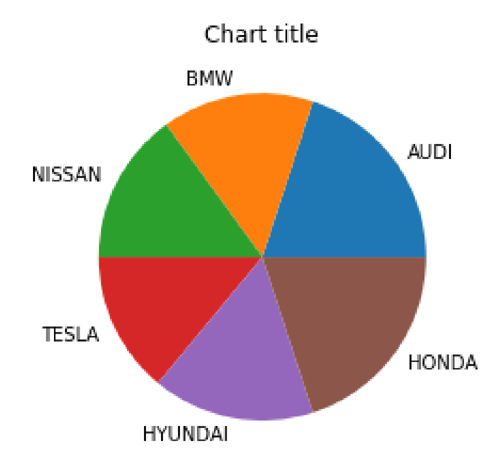
- عندما تقوم بإظهار النسب والنسب المئوية لمجموعة البيانات.
- من الأفضل استخدامه مع مجموعات البيانات الصغيرة.
- عند مقارنة تأثير عامل واحد على فئات مختلفة.
- إذا كان لديك ما يصل إلى 6 فئات.
- عندما تكون بياناتك اسمية وليست متسلسلة.

متى لا نستخدم الرسم البياني الدائري؟

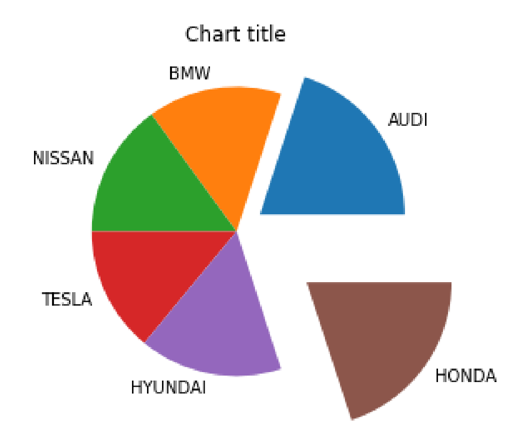
- إذا كان لديك مجموعة كبيرة من البيانات.
- إذا كنت تريد إجراء مقارنة دقيقة أو مطلقة بين القيم.

الرسم البياني الدائري باستخدام Matplotlib

```
#Creating the dataset
cars = ['AUDI', 'BMW', 'NISSAN',
        'TESLA', 'HYUNDAI', 'HONDA']
data = [20, 15, 15, 14, 16, 20]
#Creating the pie chart
plt.pie(data, labels = cars)
#Adding the aesthetics
plt.title('Chart title')
#Show the plot
plt.show()
```




```
#Creating the dataset
cars = ['AUDI', 'BMW', 'NISSAN',
        'TESLA', 'HYUNDAI', 'HONDA']
data = [20, 15, 15, 14, 16, 20]
myexplode = [0.2, 0, 0, 0,0,0.6]
#Creating the pie chart
plt.pie(data, labels = cars,explode = myexplode)
#Adding the aesthetics
plt.title('Chart title')
#Show the plot
plt.show()
```

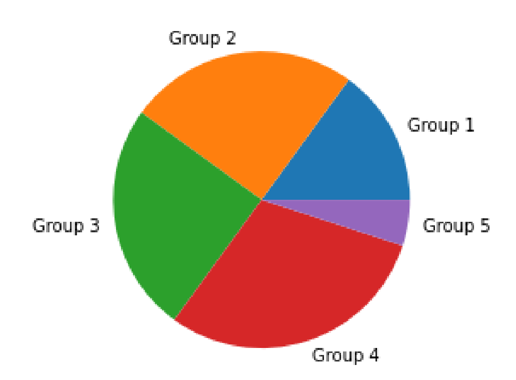


الرسم البياني الدائري باستخدام Seaborn

```
import seaborn as sns

#define data
data = [15, 25, 25, 30, 5]
labels = ['Group 1', 'Group 2', 'Group 3', 'Group 4', 'Group 5']

#create pie chart
plt.pie(data, labels = labels)
plt.show()
```



الرسم البيانى النقطى

المخطط النقطي او مخطط التشتت هو رسم بياني يوضح العلاقة بين متغيرين كنقطة على نظام إحداثيات مستطيل. يتم تحديد موضع النقطة بواسطة قيمة المتغير. من خلال مراقبة توزيع نقاط

البيانات ، يمكن استنتاج العلاقة بين المتغيرات. يتطلب إنشاء مخطط نقطي الكثير من البيانات، وإلا فإن الارتباط ليس واضحًا.

متى نستخدم الرسم البياني النقطي ؟

- لإظهار الارتباط والتجميع في مجموعات كبيرة من البيانات.
- إذا كانت مجموعة البيانات الخاصة بك تحتوي على نقاط لها زوج قيم.
- عندما تحتاج إلى ملاحظة وإظهار العلاقات بين متغيرين عدديين.
- إذا كان ترتيب النقاط في مجموعة البيانات غير ضروري.

متى لا نستخدم الرسم البياني النقطي ؟

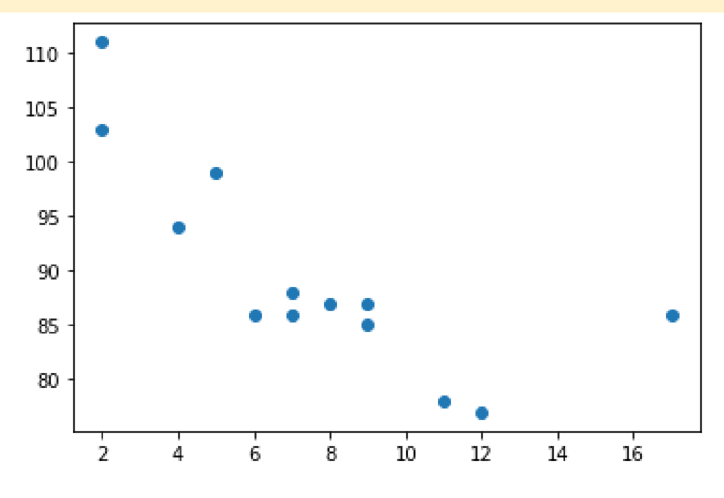
- إذا كان لديك مجموعة بيانات صغيرة.
- إذا كانت القيم في مجموعة البيانات الخاصة بك غير مرتبطة.

الرسم البياني النقطي باستخدام Matplotlib

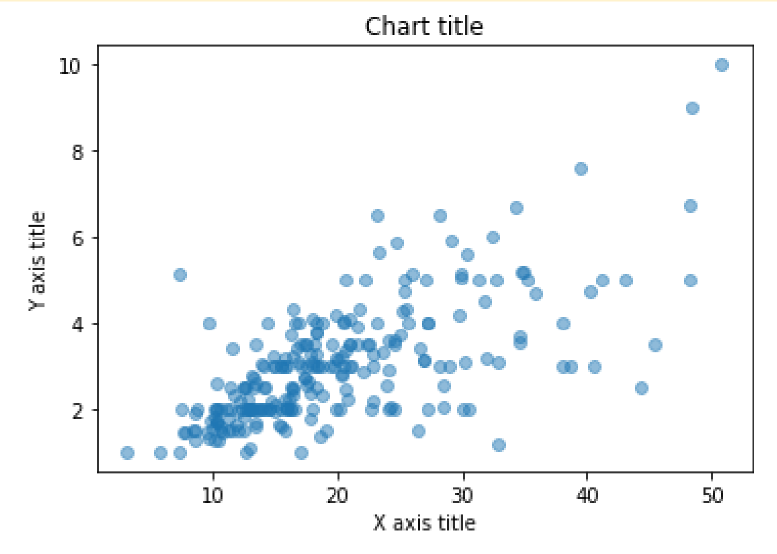
```
import matplotlib.pyplot as plt
import numpy as np

#define data
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

#Show the plot
plt.scatter(x, y)
plt.show()
```

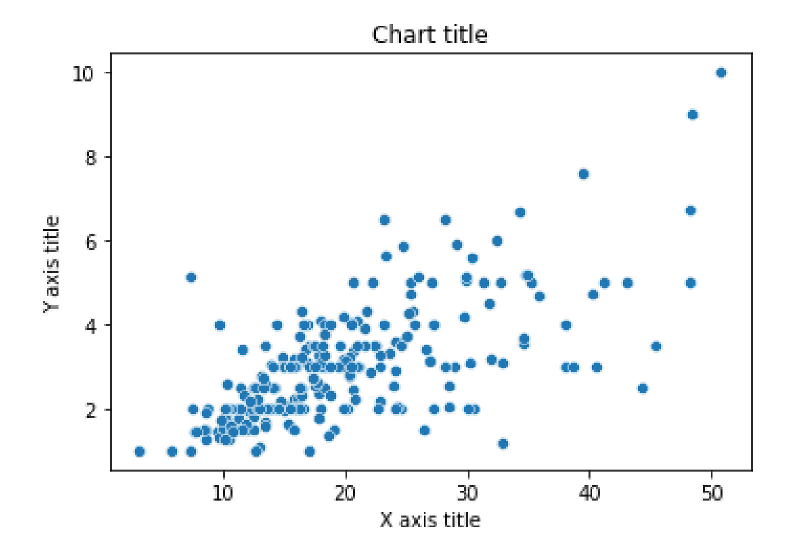


```
#Creating the dataset
df = sns.load_dataset("tips")
#Creating the scatter plot
plt.scatter(df['total_bill'],df['tip'],alpha=0.5 )
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
#Show the plot
plt.show()
```



الرسم البياني النقطي باستخدام Seaborn

```
#Creating the dataset
bill_dataframe = sns.load_dataset("tips")
#Creating scatter plot
sns.scatterplot(data=bill_dataframe, x="total_bill", y="tip")
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```



مخطط المساحة[1]

تُستخدم مخططات المساحة لتعقب التغييرات بمرور الوقت لمجموعة واحدة أو أكثر. عندما نريد تسجيل التغييرات مع مرور الوقت لأكثر من مجموعة واحدة ، يُفضل المخططات المساحية على المخططات الخطية.

متى نستخدم مخطط المساحة؟

- عندما لا ترغب في اتباع القيم بأكملها فحسب ، بل تريد أيضًا معرفة كيفية تقسيمها إلى مجموعات.
- إذا كنت ترغب في التمثيل المرئي لحجم البيانات الخاصة بك وليس فقط بمرور الوقت.

متى لا نستخدم مخططات المساحة؟

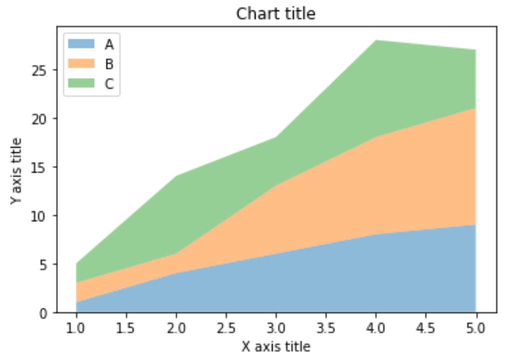
- لا يمكن استخدامه مع البيانات المتقطعة.

مخطط المساحة باستخدام Matplotlib

```
#Reading the dataset
x=range(1,6)
y=[ [1,4,6,8,9], [2,2,7,10,12], [2,8,5,10,6] ]
#Creating the area chart
ax = plt.gca()
ax.stackplot(x, y, labels=['A','B','C'],alpha=0.5)
```

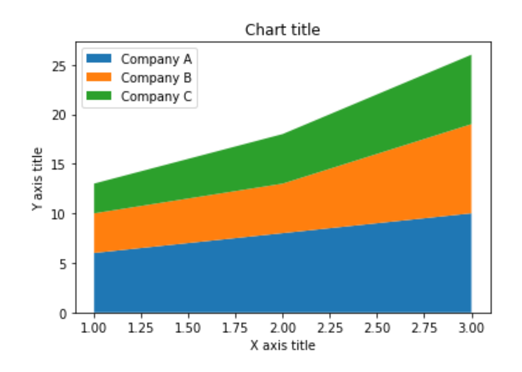
[1] Area Chart

```
#Adding the aesthetics
plt.legend(loc='upper left')
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
#Show the plot
plt.show()
```



مخطط المساحة باستخدام Seaborn

```
# Data
years_of_experience =[1,2,3]
salary=[ [6,8,10], [4,5,9], [3,5,7] ]
# Plot
plt.stackplot(years_of_experience,salary, labels=['Company A','C
ompany B','Company C'])
plt.legend(loc='upper left')
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```



المخطط الفقاعي[1]

المخطط الفقاعي هو مخطط متعدد المتغيرات ونوع من مخططات التشتت الذي يستخدم لتمثيل العلاقات بين ثلاثة متغيرات. يتم عرض قيم المتغيرات لكل نقطة بالموقع الأفقي والموضع العمودي وحجم النقطة.

متى نستخدم المخطط الفقاعي؟

- إذا كنت تريد مقارنة القيم المستقلة.
- إذا كنت تريد إظهار التوزيع أو العلاقة.
- عندما تريد رسم العلاقات بين المتغيرات الثلاثة وإظهارها.

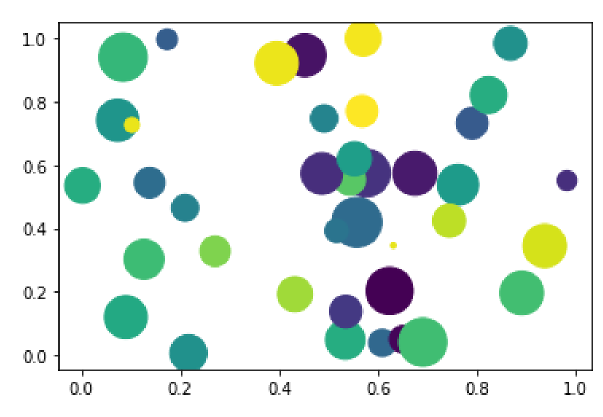
متى لا نستخدم المخطط الفقاعي؟

- إذا كان لديك مجموعة بيانات صغيرة.

المخطط الفقاعي باستخدام Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np

# create data
x = np.random.rand(40)
y = np.random.rand(40)
z = np.random.rand(40)
colors = np.random.rand(40)
# use the scatter function
plt.scatter(x, y, s=z*1000,c=colors)
plt.show()
```



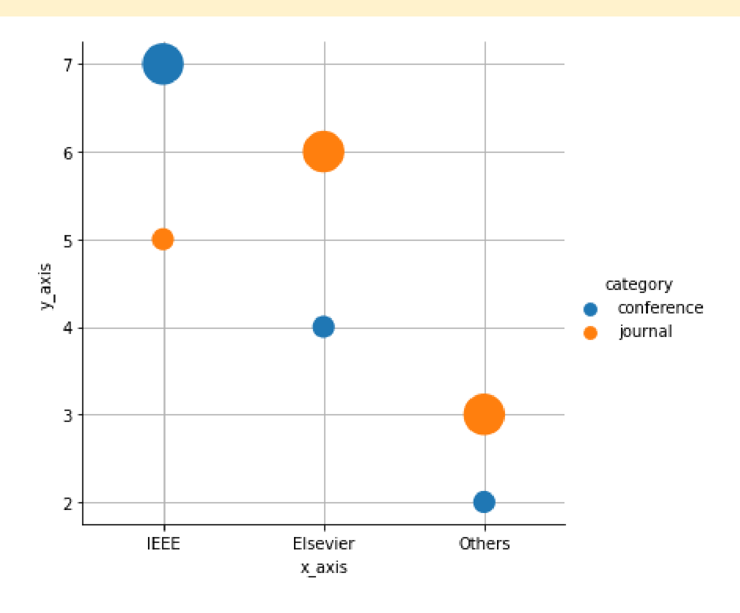
[1] Bubble Chart

المخطط الفقاعي باستخدام Seaborn

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pandas as pd

# data
x=["IEEE", "Elsevier", "Others", "IEEE", "Elsevier", "Others"]
y=[7, 6, 2, 5, 4, 3]
z=["conference", "journal", "conference", "journal", "conference", "
journal"]

# create pandas dataframe
data_list = pd.DataFrame(
    {'x_axis': x,
     'y_axis': y,
     'category': z
    })
# change size of data points
minsize = min(data_list['y_axis'])
maxsize = max(data_list['y_axis'])
# scatter plot
sns.catplot(x="x_axis", y="y_axis", kind="swarm", hue="category
",sizes=(minsize*100, maxsize*100), data=data_list)
plt.grid()
```



خلاصة الفصل الثالث

- البيانات هي محرك يمكنه توجيه الأعمال في الاتجاه الصحيح.
- غالبًا يمكن اعتبار مجموعة البيانات على أنها مجموعة كائنات بيانات لها نفس الخصائص.
- تتضمن الاسماء الأخرى لكائن البيانات: سجل ، أو نقطة ، أو متجه ، أو نمط ، أو حدث ، أو عنصر ، أو عينة ، أو عرض ، أو كيان.
- تنظيف البيانات عملية إعداد البيانات للتحليل عن طريق إزالة أو تعديل البيانات غير الصحيحة أو غير الكاملة أو غير الملائمة أو المكررة أو غير المناسبة.
- يعتبر تنظيف البيانات عنصرًا رئيسياً في أساسيات علم البيانات.
- يمكن لخوارزمية بسيطة التغلب على خوارزمية معقدة لمجرد أنها تحصل على بيانات كافية وعالية الجودة.
- جودة البيانات لا تقل أهمية عن جودة النموذج أو التصنيف التنبئي.
- لا يمكن اعتبار البيانات المتطرفة ضوضاء أو خطأ. ومع ذلك ، يشتبه في أنها لم يتم إنتاجها بنفس طريقة إنتاج البيانات (الكائنات) الأخرى.
- البيانات المتطرفة لها تأثير كبير على نتيجة تحليل البيانات.
- يمكنك فقط حذف جزء من البيانات (سمة) غير ذي صلة. ربما تكون الميزة التي تبدو غير ملائمة بالنسبة لك وثيقة الصلة بمجال مثل منظور إكلينيكي.
- تجميع البيانات هو طريقة يتم فيها جمع البيانات الأولية واستخدامها بإيجاز للتحليل.
- يسمى تغيير البيانات من القيم المستمرة إلى القيم المنفصلة بالتقطيع.
- يتم تنعيم البيانات باستخدام خوارزميات متخصصة لإزالة الضوضاء من مجموعة البيانات.
- في بعض الأحيان ، قد يؤدي تنعيم البيانات إلى إزالة نقاط البيانات القابلة للاستخدام.
- أكثر تقنيات التحجيم شيوعًا هي التوحيد القياسي والتسوية.
- يتقارب الانحدار الاشتقاقي للشبكة العصبية مع ميزة التحجيم بشكل أسرع بكثير من بدونها.
- يحدث التوحيد عندما تختلف خصائص مجموعة بيانات الإدخال اختلافًا كبيرًا في نطاقها.
- تعد التسوية مفيدة عندما تعلم أن توزيع البيانات الخاص بك لا يتبع التوزيع الغاوسي (منحنى الجرس).

مصادر اضافية لمزيد من القراءة

Boschetti, A., & Massaron, L. (2015). Python data science essentials. Packt Publishing Ltd.

Brownlee, J. (2020). Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python. Machine Learning Mastery.

Kazil, J., & Jarmul, K. (2016). Data wrangling with Python: tips and tools to make your life easier. " O'Reilly Media, Inc.".

McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".

Rattenbury, T., Hellerstein, J. M., Heer, J., Kandel, S., & Carreras, C. (2017). Principles of data wrangling: Practical techniques for data preparation. " O'Reilly Media, Inc.".

Rogel-Salazar, J. (2020). Advanced Data Science and Analytics with Python. Chapman and Hall/CRC.

Shah, C. (2020). A hands-on introduction to data science. Cambridge University Press.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). Data mining introduction.

الجزء الثاني

تعلم الآلة

يشمل الفصول التالية:

الفصل الرابع: مقدمة في تعلم الآلة.

الفصل الخامس: اختيار النموذج وتقييمه.

الفصل السادس: التعلم الخاضع للاشراف.

الفصل السابع: التعلم العميق.

الفصل الثامن: التعلم غير الخاضع للاشراف.

الفصل التاسع: موضوعات مختارة.

4

مقدمة فى تعلم الآلة

الأهداف:

- ما هو التعلم الآلي؟
- التعرف على أنواع مختلفة من مناهج التعلم.
- العلاقة بين التعلم الآلي والتخصصات الأخرى.
- تطبيق التعلم الآلي.
- التعرف على أدوات التعلم الآلي.
- التعرف على قابلية التفسير والتوضيح.

ما هو التعلم الآلي؟

بعد المطر، أمشيفي شارع مبتل. مع الشعور بالنسيم اللطيف ورؤية توهج غروب الشمس، نراهن أن الطقس يجب أن يكون جيداً غداً. ما زلنا نتوقع الحصول على درجات جيدة هذا الفصل الدراسي بعد العمل الجاد والدراسة الجادة. بإلقاء نظرة فاحصة على هذه الأمثلة، نجد أن العديد من توقعاتنا تستند إلى الخبرة. على سبيل المثال، لماذا نتوقع طقسًا جيدًا غدًا بعد رؤية نسيم لطيف وشرق الشمس؟ نتطلع إلى هذا الطقس الجيد لأننا نتمتع بتجربة أن الطقسفي اليوم التالي غالبًا ما يكون جيدًا عندما نشهد مثل هذا المشهد الآن. وبالمثل، تخبرنا تجربة التعلم لدينا أن العمل الجاد يؤدي إلى درجات أكاديمية جيدة. نحن واثقونفي توقعاتنا، لأننا تعلمنا من التجربة واتخذنا قرارات بناءً على التجربة. بينما يتعلم البشر من التجربة، هل يمكن لأجهزة الكمبيوتر أن تفعل الشيء نفسه؟ الجواب نعم، والتعلم الآلي هو ما يفعله. التعلم الآلي هو طريقة تعمل على تحسين أداء النظام من خلال التعلم التجريبي من خلال الأساليب الحسابية (Zhou، 2021).

صاغ آرثر صموئيل، رائد ألعاب الكمبيوتر والذكاء الاصطناعي، مصطلح "التعلم الآلي"في عام 1959في شركة IBM. ووصف التعلم الآلي بأنه *"السياق الذي يمكن أن تتعلم فيه أجهزة الكمبيوتر بدون برمجة واضحة"*. التعلم الآلي هو فرع من فروع الذكاء الاصطناعي تم إنشاؤه لإنشاء أجهزة كمبيوتر لديها القدرة على التعلم. الفكرة الرئيسية لاختراع التعلم الآلي القائم على العينة هو أن عملية التفكيرفي مشكلة معينة ممكنة من خلال الإشارة إلى أمثلة سابقة مماثلة. الأمثلة السابقة المستخدمة لبناء القدرات تسمى أمثلة التدريب (البيانات) وعملية القيام بذلك تسمى التعلم. ولكن قبل أن نتعمق أكثرفي التعلم الآلي، دعونا نلقي نظرة على ماهية التعلم. يمكن تعريف التعلم بأنه "تحسين الأداءفي مهمة معينة باستخدام الخبرة والممارسة". يحدث السلوك الذكي للإنسان من خلال التعلمفي التجارب، والتعلم هو خالق المرونةفي الحياة الفردية. التعلم البشري هو شيء يتجاوز بل وأكثر تقدمًا من خوارزميات التعلم الآلي الأكثر تقدمًا. ومع ذلك، كيف يحدث التعلم على أجهزة الكمبيوتر؟في أنظمة الكمبيوتر، توجد الخبرةفي شكل بيانات، وتتمثل المهمة الرئيسية للتعلم الآليفي تطوير خوارزميات التعلم التي تقوم بنمذجة البيانات. من خلال تغذية البيانات التجريبية إلى خوارزمية التعلم الآلي، نحصل على نموذج يمكنه عمل تنبؤاتفي الملاحظات الجديدة.

تعريف التعلم

"سيكون برنامج الكمبيوترقادراً على التعلم من خلال النظرفي تجربة E للمهمة T وفقاًالمعيارالأداء P، إذا تحسن أدائه بعد تجربة E للمهمة T."

امثلة:

- **تعلم التعرف على خط اليد**
  - المهمة T: التعرف على الكلمات المكتوبة بخط اليد فى الصور وتصنيفها.
  - الاداء P: النسبة المئوية للكلمات التى تم تصنيفها بشكل صحيح.
  - تجربة E: مجموعة من الكلمات المكتوبة بخط اليد مع تصنيفات معينة.
- **تعلم الروبوت السائق**
  - المهمة T: القيادة على الطرق السريعة باستخدام المستشعرات البصرية.
  - الأداء P: متوسط المسافة المقطوعة قبل الخطأ.
  - تجربة E: سلسلة من الصور والأوامر عندما يقودها سائق بشرى
- **تعلم الشطرنج**
  - المهمة T: لعب الشطرنج.
  - الكفاءة P: نسبة الفوز بالمباريات ضد الخصوم.
  - تجربة E: ممارسة الألعاب ضد نفسه

تبدأ عملية التعلم الآلي باستخدام البيانات الأولية لاستخراج معلومات مفيدة للمساعدة في اتخاذ قرارات أفضل. وبالتالي ، يمكن تعريف التعلم الآلي على أنه: "*يركز التعلم الآلي على تصميم النماذج التي تكون فيها خوارزمية كمبيوتر تعتمد على بيانات التدريب المعطاة لنموذج التعلم تلقائيًا في منطقة معينة" تحصل تلقائيًا من خلال التجربة والاختبار على البيانات ويحسن أداءه حتى يتصرف بشكل مشابه للبشر في مواجهة البيانات الجديدة في نفس المجال* ". على الرغم من أننا قدمنا تعريفًا عامًا للتعلم الآلي ، يقدم مؤلفون مختلفون تعريفات مختلفة للتعلم الآلي. فيما يلي ثلاثة تعريفات أخرى للتعلم الآلي:

- *التعلم الآلي هو برمجة أجهزة الكمبيوتر لتحسين مقاييس الأداء باستخدام بيانات نموذجية أو تجربة سابقة. لدينا نموذج محدد لبعض المعاملات ، والتعلم هو تنفيذ برنامج كمبيوتر لتحسين معاملات النموذج باستخدام البيانات التعليمية أو الخبرة السابقة. قد يكون هذا النموذج تنبؤيًا لعمل تنبؤات في المستقبل ، أو لاكتساب المعرفة من البيانات ، أو كليهما. (Alpaydin, 2004)*
- *يهتم مجال البحث المعروف باسم التعلم الآلي بكيفية إنشاء برامج الكمبيوتر التي يتم تحسينها تلقائيًا من خلال التجربة. (Mitchell, 1997)*

- *يعني التعلم الآلي أن أجهزة الكمبيوتر تعدل أو تكيف إجراءاتها (سواء كانت تنبؤية أو تتحكم في روبوت) لجعلها أكثر دقة ، حيث تنعكس الدقة في مقدار الإجراءات المحددة بشكل صحيح. (Marsland, 2015)*

وفقًا لهذه التعريفات ، يطرح التعلم الآلي سؤالين أساسيين: أولاً ، كيف يعرف الكمبيوتر أنه يعمل على تحسين أدائه في مهمة معينة؟ ثانيًا ، كيف يفهم البرنامج أنه بحاجة إلى تحسين في هذه المهمة؟ توفر الإجابات على هذه الأسئلة تصنيفًا للعديد من الأساليب المختلفة للتعلم الآلي، والتي سنصفها في هذا الفصل تحت عناوين التعلم الخاضع للإشراف ، وغير الخاضع للإشراف، والتعلم المعزز.

غالبًا ما يتم تقسيم أساليب التعلم الآلي إلى مرحلتين (*Hertzmann and Fleet, 2012*):

1. **التعليم**: نموذج يحقق التعلم باستخدام مجموعات البيانات التعليمية.
2. **التطبيق**: يستخدم النموذج المدرب لاتخاذ القرارات والتنبؤات على بعض البيانات التجريبية (البيانات التي لم يلاحظها النموذج في عملية التدريب).

التعلم الخاضع للاشراف

في طريقة التعلم الخاضع للاشراف، يتم تغذية مجموعة من عينات التدريب للخوارزمية بالإجابات الصحيحة (الأهداف) (لدينا بيانات إدخال أولية ونتائجها) وتحاول الخوارزمية تعلم دالة بناءً على هذه البيانات والإجابات الصحيحة (يتعلم بمرور الوقت ويصبح أكثر دقة بمرور الوقت) لتكون قادرة على التنبؤ بدقة بالقيم المستهدفة للعينات الجديدة. بمعنى آخر، الهدف هو تكييف النظام بطريقة تمكن النظام من التنبؤ بالمخرجات الصحيحة للمدخلات الجديدة بناءً على ما تعلمه حتى الآن من بيانات التدريب. يسمى هذا النوع من طريقة التعلم أيضًا التعلم من الأمثلة.في التعلم الخاض للاشراف ، إذا كانت بيانات مشكلة التعلم منفصلة ، يتم تصنيفها ، وإذا كانت قيم البيانات مستمرة ، فإنها تسمى الانحدار (التوقع).

تعريف التعلم الخاضع للاشراف

التعلم الخاضع للأشراف هو المكان الذى يكون لديك فيه متغيرات الإدخال (X) ومتغير الإخراج (Y) واستخدام خوارزمية لتعلم دالة التطبيق من الإدخال إلى الإخراج:

$$Y = f(X)$$

الهدف هو تقدير دالة التطبيق جيداً بحيث عندما يتم توفير بيانات إدخال جديدة (X) للخوارزمية، يمكنها التنبؤ بمتغيرات الإخراج (Y) لتلك البيانات.

نظراً لأن عملية تعلم الخوارزمية من مجموعة البيانات التعليمية يمكن اعتبارها معلما يشرف على عملية التعلم، فإنها تسمى التعلم الخاضع للأشراف. نحن نعرف الإجابات الصحيحة، تقوم الخوارزمية بشكل متكرر بعمل تنبؤات حول البيانات التعليمية ويتم تصحيحها من قبل المعلم. يتوقف التعلم عندما تصل الخوارزمية إلى مستوى مقبول من الأداء.

فوائد التعلم الخاضع للاشراف

- التعلم الخاضع للأشراف مفيد لاستخلاص النتائج من الخبرة السابقة أو المعرفة السابقة.
- يمكنه حل أنواع مختلفة من المشاكل الحسابية العملية.
- النتيجة أدق من طريقة التعلم غير الخاضع للاشراف.
- قبل تقديم البيانات للتدريب ، أنت تعرف بالضبط عدد الفئات الموجودة.

مساوئ التعلم الخاضع للاشراف

- يستغرق التدريب وقتًا طويلاً في الحساب.
- إذا كانت مجموعة البيانات التجريبية مختلفة عن مجموعة بيانات التدريب ، فإن خوارزمية التعلم الخاضع للاشراف ستواجه مشاكل عند توقع النتائج.
- يعد وضع علامات على البيانات مضيعة للوقت ومكلفًا ، وفي بعض الأحيان لا يمكن جمع مجموعة بيانات مصنفة بشكل كافٍ.
- التعلم الخاضع للاشراف محدود للغاية بحيث لا يمكن تنفيذ بعض مهام التعلم الآلي المعقدة باستخدامه.
- لا يمكن أن يوفر التعلم الخاضع للاشراف معلومات غير معروفة من البيانات التعليمية مثل التعلم غير خاضع للاشراف.
- إذا كانت مسألة تصنيف ، إذا قدمنا مدخلات ليست جزءًا من أي من الفئات في مجموعة بيانات التدريب ، فقد يكون ناتج تسمية الفئة غير صحيح. على سبيل المثال ، افترض أنك قمت بتدريب مصنف الصور باستخدام بيانات القطط والكلاب. ثم إذا أعطيت صورة زرافة ، فقد يكون الناتج قطة أو كلبًا ، وهذا غير صحيح.

التعلم غير خاضع للاشراف

في النهج غير خاضع للاشراف ، لا يتم تقديم الإجابات الصحيحة للخوارزمية (لم يتم تصنيف البيانات) ، ولكن بدلاً من ذلك تحاول الخوارزمية تحديد أوجه التشابه بين المدخلات بحيث يتم تجميع المدخلات التي لها ميزة مشتركة معًا. بمعنى آخر ، لا يحتوي النظام على المخرجات المناسبة ، لكنه يستكشف البيانات ويمكنه استخلاص استنتاجات من مجموعة البيانات لوصف الهياكل المخفية للبيانات غير المسماة. التجميع وقواعد الارتباط وتقليل الأبعاد هي أمثلة على التعلم غير الخاضع للإشراف.

تعريف التعلم غير الخاضع للاشراف

التعلم غير الخاضع للإشراف هو المكان الذى يكون لديك فيه بيانات (X) فقط ولا توجد متغيرات إخراج مقابلةً.

الهدف من التعلم غير الخاضع للإشراف هو معرفةُ المزيد من البيانات.

وهذا ما يسمى التعلم غير الإشرافى لأنه ، على عكس التعلم الإشرافى ، لا توجد إجابات صحيحة ولا يوجد معلم. تقع على عاتق الخوارزميات مسؤولية اكتشاف وتقديم بنية مثيرة للاهتمام فى البيانات.

مع الخوارزميات غير الخاضعة للاشراف ، لا تعرف ما الذى تريد الخروج منه من النموذج. قد تشك فى أنه يجب أن يكون هناك نوع من العلاقة أو الارتباط بين بياناتك، لكن البيانات معقدة للغاية بحيث لا يمكن تخمينها. لذلك فى هذه الحالات، تقوم بتحويل بياناتك إلى بيانات عادية لجعلها قابلة للمقارنة ثم ترك النموذج يعمل ومحاولة العثور على بعض هذه العلاقات. تتمثل إحدى الميزات الخاصة لهذه النماذج فى أنه بينما يمكن للنموذج اقتراح طرق مختلفة لتجميع بياناتك أو طلبها، فإن الأمر متروك لك لإجراء المزيد من البحث حول هذه النماذج للكشف عن شيء مفيد.

فوائد التعلم غير الخاضع للاشراف

- يتطلب وضع العلامات على البيانات الكثير من العمل والمال. التعلم غير الخاضع للإشراف يحل هذه المشكلة من خلال التعلم من البيانات غير المسماة (غير معلمة).
- إنه مفيد جدًا في العثور على أنماط البيانات التي لا يمكن العثور عليها باستخدام الطرق التقليدية.
- يتم تقليل حجم البيانات بسهولة باستخدام هذا النوع من التعلم.

مساوئ التعلم غير الخاضع للاشراف

- قد تكون النتيجة أقل دقة من طريقة التعلم الإشرافي. لأننا لا نملك أي تسميات للبيانات ويجب أن يتعلم النموذج بالمعرفة المكتسبة من البيانات الأولية.
- كلما زادت الميزات ، زادت تعقيدها.
- إنها عملية تستغرق وقتًا طويلاً. لأن مرحلة تعلم الخوارزمية قد تستغرق الكثير من الوقت لتحليل وحساب جميع الاحتمالات.

التعلم المعزز

في التعلم المعزز، يحاول الوكيل حل مشكلة عن طريق التجربة والخطأ من خلال التفاعل مع بيئة طبيعتها غير معروفة للوكيل. يمكن للوكيل تغيير حالة البيئة من خلال أفعاله أثناء تلقي تعليقات فورية من البيئة. هدف الوكيل هو حل المشكلة من خلال إيجاد سلسلة الإجراءات المثلى. على الرغم من أن التعلم المعزز هو أحد مجالات التعلم الآلي، إلا أنه يختلف اختلافًا جوهريًا عن أساليب التعلم الآلي القياسية (الاشرافي وغير الاشرافي) من نواح كثيرة. أولاً، لا يعتمد التعلم المعزز على تعلم البيانات. بدلاً من ذلك،في التعلم المعزز، يتعلم الوكيل من خبرته المكتسبة أثناء التفاعل مع البيئة ولا يعتمد على المشرف. ثانيًا، يركز التعلم المعزز على إيجاد السياسة المثلى بدلاً من تحليل البيانات.

الفرق بين التعلم الاشرافي وغير الاشرافي والمعزز

يحدث التعلم الإشرافي عندما يتم تغذية الخوارزمية بمجموعة من الأمثلة التعليمية بالإجابات الصحيحة (الأهداف) وتستخدم الخوارزمية هذه الإجابات كدليل للمساعدة في حل المشكلة. في المقابل ، لا يتطلب التعلم غير الإشرافي بيانات معنونة ، والنموذج نفسه هو الذي يحل المشكلة بمفرده ، بدون مراقب خارجي ، من خلال إيجاد أوجه التشابه بين المدخلات واكتشاف الأنماط المخفية. على عكس هذين النهجين ، لا يتطلب التعلم المعزز مجموعة بيانات ويتفاعل الجهاز أو الوكيل مع بيئته للحصول على أفضل إجراء لحل المشكلة عن طريق التجربة والخطأ وتلقي المكافآت من البيئة. التعلم المعزز ليس اشرافياً تمامًا ، لأنه لا يعتمد كليًا على مجموعات بيانات التدريب (المسماة). في الواقع ، يعتمد التعلم المعزز على القدرة على مراقبة رد الفعل على الإجراءات المتخذة وقياسها بالمكافأة. كما أنها ليست غير إشرافية ، لأننا نعرف بالفعل متى نمثل "المتعلم" ، وهي المكافأة المتوقعة. باختصار ، في التعلم الخاضع للاشراف ، الهدف هو إنشاء صيغة تعتمد على قيم المدخلات والمخرجات. في التعلم غير الخاضع للإشراف ، تم العثور على اتصال بين قيم الإدخال وتجميعها. في التعلم المعزز ، يتعلم الوكيل من خلال التفاعل

مع البيئة. بناءً على ذلك ، يمكن رؤية الفرق بين هذه الأساليب الثلاثة للتعلم الآلي في الجدول 4-1.

جدول 4ـ1 مقارنة بين التعلم الاشرافي وغير الاشرافي والمعزز

	التعلم الاشرافي	التعلم غير الاشرافي	التعلم المعزز
التعريف	يتعلم من خلال مجموعات البيانات ذات العلامات.	بدون توجيه ، يتم تدريبه من خلال البيانات غير المسماة.	يتفاعل مع بيئة العمل.
نوع البيانات	البيانات المصنفة	البيانات غير مصنفة	لا يوجد تعريف للبيانات
نوع المسألة	التصنيف والتوقع	قواعد الارتباط والتجميع	على أساس المكافآت
المشرف	مشرف اضافي	بدون مشرف	بدون مشرف
الهدف	تعيين بيانات الإدخال لمخرجات محددة	اكتشاف النمط	تعلم سلسلة من الإجراءات

التعلم الانتقالي

يركز التعلم الانتقالي على استخراج البيانات من مجال مشابه لزيادة القدرة على التعلم أو تقليل عدد العينات ذات العلامات المطلوبةفي المجال المستهدف.في التعلم الانتقالي، يستخدم أحد النماذج المعرفة المكتسبة من العمل السابق لتحسين التعميم على نموذج آخر. الغرض من التعلم الانتقالي هو تحسين عملية تعلم المهام الجديدة باستخدام الخبرة المكتسبة من حل المشكلات السابقة المتشابهة إلى حد ما.

تعريف التعلم الانتقالي

استخدام نموذج مدرب مسبقًا لنقل المعرفة من هذا النموذج إلى مهمة مماثلة، من أجل تحسين أداء هذه المهمة الجديدة.

لفهم التعريف الرسمي للتعلم الانتقالي، من الضروري أولاً تحديد المجال والدالة. المجال عبارة عن مجموعة من البيانات التي يتم استخدامها لتعليم الاستخدام ويتم تمثيل المجال على أنه $D = \{\chi, P(X)\}$ والذي يتكون من مكونين: χ مساحة الميزة و$P(X)$ توزيع احتمالي في هذا التعريف$X = \{x_1, \ldots, x_n\} \epsilon \chi$. يمكن تمثيل الدالة بمساحة التسمية y ودالة النموذج المستهدف$f(x)$ وعلى النحو $T = \{y, f(x)\}$ ..يمكن أيضًا كتابة $f(x)$كدالة احتمالية مشروطة. يمكن الآن تعريف التعلم الانتقالي رسميًا على النحو التالي:

بالنظر إلى المجال المصدر D_s ودالة التعلم المصدر T_s، والمجال الهدف D_t ، ودالة التعلم الهدف T_t ، حيث يكون حجم D_s أكبر من حجم D_t ، يعد التعلم الانتقالي طريقة لتحسين أداء النموذج الهدف $f_T(.)$للدالة التعلم بهدف T_t. مع الحصول على معرفة ضمنية من D_sو D_t ، حيث $D_s \neq D_t$ و $T_s \neq T_t$.

التعلم متعدد المهام

يعد تعدد المهام نموذجًا تعليميًا يتم من خلاله تعلم العديد من المهام ذات الصلةفي وقت واحد من أجل زيادة كفاءة النموذج إلى الحد الأقصى، كما يتم تحسين العديد من دوال الخطأفي وقت واحد.في هذه العملية، يستخدم النموذج جميع البياناتفي مهام مختلفة لمعرفة التمثيل العام للبيانات المفيدةفي سياقات مختلفة ومشاركتها بين المهام المختلفة. تزيد هذه المشاركات من كفاءة النموذج (يمكن أن تستفيد كل مهمة من مهمة أخرى) ويمكن أن تؤدي إلى سرعات تعلم أسرع. الدافع الرئيسي للتعلم متعدد المهام هو إنشاء نموذج "عام" يمكنه حل مهام متعددةفي وقت واحدفي نموذج واحد بدلاً من إنشاء عدة نماذج "متخصصة" يتم تدريبها فقط لمهمة محددة. من وجهة نظر بيولوجية، فإن تعدد المهام مستوحى من الطريقة التي نتعلم بها نحن البشر. لتعلم مهام جديدة، عادة ما نطبق المعرفة التي اكتسبناها من تعلم المهام ذات الصلة.

الفرق بين التعلم الانتقالي وتعلم تعدد المهام

يختلف التعلم متعدد المهام من التعلم الانتقالي ، ويختلف في كيفية نقل المعرفة. يتم تعلم المهام بالتسلسل في التعلم الانتقالي ونقلها من واحد إلى آخر. في حين أن التعلم متعدد المهام من خلال مشاركة المعلومات بين جميع المهام يؤدي إلى أداء جيد في جميع المهام التي ينظر فيها نموذج واحد في وقت واحد.

التعلم مع عينة واحدة[1]

عادةً ما يتضمن تصنيف العملية تغذية عدد كبير من العينات من كل فئة إلى النموذج. التعلم بأحد العينات على عكس نماذج التعلم الآلي التقليدية ، التي تستخدم آلاف العينات التدريبية للتعلم ، فهو نوع من التعلم يستخدم واحدًا فقط أو عددًا قليلاً من العينات التعليمية للتعلم. مثال على تطبيق لهذا النوع من التعلم هو التعرف على الوجوه. حيث يجب تصنيف الأشخاص بشكل صحيح وفقًا لتعبيرات الوجه المختلفة وظروف الإضاءة والإكسسوارات وتسريحات الشعر وفقًا لصورة نمطية واحدة أو أكثر.

التعلم بدون عينات [2]

تركز العديد من أساليب التعلم الآلي اليوم على تصنيف العناصر التي تم تدريب فئاتها بالفعل. ومع ذلك ، من الضروري في كثير من الحالات تصنيف العناصر التي لم يتم رؤية فئاتها من قبل.

---

[1] One-shot Learning

[2] Zero-shot learning

تعلم بدون عينات هو طريقة تعلم إشرافية ولكن بدون بيانات تعليمية من هذه الفئة. التعلم بدون عينات قادر على حل مشكلة ما دون تلقي أي تعليمات إرشادية من تلك المشكلة. على سبيل المثال ، تخيل أنك تعرفت على مجموعة من الكائنات في صورة ما دون أن تكون قد رأيت مسبقًا صورة لهذا النوع من الأشياء.

التعلم الاستقرائي [1] (التعلم المفهومي [2])

يتضمن التعلم الاستقرائي ، المعروف أيضًا باسم التعلم المفهومي، إنشاء قاعدة عامة لجميع البيانات التي يتم تغذيتها في الخوارزمية. في هذا النهج ، لدينا البيانات كمدخلات والنتائج كمخرجات ، ونحتاج إلى إيجاد العلاقة بين المدخلات والمخرجات. يمكن أن يكون هذا معقداً للغاية اعتمادًا على البيانات. ومع ذلك ، فهي طريقة فعالة تستخدم في التعلم الآلي في مختلف المجالات مثل تقنية التعرف على الوجوه وتشخيص الأمراض والعلاج وما إلى ذلك. هذا النوع من التعلم هو نهج من أسفل إلى أعلى.

في هذا النوع من التعلم ، يتم إنشاء النموذج بطريقة أنه إذا كان بإمكانه تمديد تقريب جيد للدالة الموضوعية لمجموعة كبيرة من المجموعات التعليمية ، فيمكنه تقدير هذه الدالة الموضوعية في حالة الأمثلة غير المرئية. في هذا التعلم ، المعلومات الوحيدة المتاحة هي مجموعة بيانات التدريب ، لذلك في أحسن الأحوال يمكن لخوارزمية التعلم أن تقدم فرضية تقدر الدالة الموضوعية في عينات التدريب.

هذا التعلم مهم جداً لأنه يعطينا علاقة بالبيانات التي يمكن استخدامها للرجوع إليها في المستقبل. يتم استخدام هذا النهج عندما لا تكون الخبرة البشرية قابلة للتطبيق عندما تتغير المخرجات. باختصار ، في التعلم الاستقرائي ، نقوم بتعميم نتائج الحقائق. علي سبيل المثال:

آ. التفاح فاكهة.

ب. طعم التفاح حلو.

النتيجة: جميع الفواكه لها طعم حلو.

لا يزال هذا المجال من التعلم الآلي قيد الدراسة والبحث. لأن هناك العديد من الاقتراحات لتحسين أداء وسرعة الخوارزمية.

التعلم الاستقرائى هو ما نشير إليه عادةً بالتعلم التقليدى. نحن نبنى ونعلم نموذجًا للتعلم الآلى بناءً على مجموعات بيانات التدريب المسماة التى لدينا بالفعل. ثم

[1] Inductive Learning

[2] Concept Learning

نستخدم هذا النموذج المدرب للتنبؤ بتسميات مجموعة البيانات التجريبية التي لم نواجهها من قبل.

التعلم الاستنتاجي [1]

تمامًا مثل التفكير الاستقرائي ، يعد التعلم الاستنتاجي أو التفكير الاستنتاجي شكلاً آخر من أشكال التفكير. في الواقع ، التفكير هو مفهوم للذكاء الاصطناعي ، وكل من التعلم الاستقرائي والاستنتاجي جزء منه. الاستدلال الاستنتاجي هو عملية استنتاج معرفة جديدة من المعلومات الموجودة الموجودة ذات الصلة منطقيًا وشكل صحيح من المنطق ، مما يعني أن الاستنتاج يجب أن يكون صحيحًا إذا كانت الادعاءات صحيحة.

على عكس التعلم الاستقرائي ، الذي يعتمد على تعميم حقائق محددة ، يستخدم التعلم الاستنتاجي الحقائق والمعلومات الموجودة للتوصل إلى استنتاجات صحيحة ويستخدم نهجًا من أعلى إلى أسفل (على عكس التعلم الاستقرائي). النقطة المهمة التي يجب ملاحظتها هي أنه في التعلم الاستنتاجي ، تكون النتائج محددة ؛ هذا هو ، إما نعم أو لا. بينما يعتمد التعلم الاستقرائي على الاحتمالات ، إلا أنه يمكن أن يختلف من قوي إلى ضعيف.

صحة[2] الاستدلال تضمن حقيقة الاستنتاج المنطقي. يبدأ التفكير الاستنتاجي عادةً بالمبادئ الأساسية ويصل إلى نتيجة محددة. علي سبيل المثال:

آ. جميع الحيوانات آكلة اللحوم تأكل اللحوم.

ب. الاسد هو آكل لحوم.

النتيجة: الاسد يأكل اللحوم.

تتبع الطريقة الكلاسيكية للتعلم الآلي النموذج العلمي للاستقراء والقياس. في المرحلة الاستقرائية، نتعلم النموذج من البيانات الأولية (المجموعة التعليمية) وفي المرحلة القياسية، يتم تطبيق النموذج للتنبؤ بسلوك البيانات الجديدة.

التعلم المجازي [3]

في التعلم المجازي ، يتم تحليل البيانات التعليمية والبيانات التجريبية مسبقًا. المعرفة المكتسبة من مجموعة البيانات هذه هي المعرفة المفيدة. يحاول هذا النموذج توقع العلامات لمجموعة البيانات التجريبية بعد التعلم من مجموعة بيانات التدريب.

---

1 Deductive Learning

2 validity

3 Transductive Learning

الفرق بين التعلم الاستقرائي والتعلم المجازي

التعلم الاستقرائي ليس سوى المبدأ الكامن وراء خوارزميات التعلم الآلي الخاضعة للإشراف حيث يحاول النموذج إنشاء علاقة بين متغيرات الميزة والمتغير المستهدف من خلال فحص الأنماط المخفية في البيانات التعليمية. على الرغم من تعرض النموذج لمجموعة محدودة من البيانات التعليمية ، فإن تعلم النموذج سيكون متسقًا مع الطبيعة العامة للبيانات بحيث يمكنه التنبؤ بقيمة كل نقطة بيانات من مجموعة بيانات غير مسماة (مجموعة بيانات تجريبية). يتمثل الاختلاف الرئيسي بين التعلم الاستقرائي والتعلم المجازي في أنه أثناء التعلم المجازي ، واجهت بالفعل مجموعات بيانات تعليمية وتجريبية عند تدريب النموذج. ومع ذلك ، يواجه التعلم الاستقرائي البيانات التعليمية فقط عند تدريب نموذج ويطبق النموذج الذي تم تعلمه على مجموعات البيانات التي لم يسبق لها مثيل من قبل.

لا يصنع التعلم المجازي نموذجًا تنبؤيًا. إذا تمت إضافة نقطة بيانات جديدة إلى مجموعة البيانات التجريبية ، فعلينا إعادة تشغيل الخوارزمية من البداية ، وتدريب النموذج ، ثم استخدامه للتنبؤ بالعلامات. من ناحية أخرى ، يشكل التعلم الاستقرائي نموذجًا تنبؤيًا. عندما تواجه نقاط بيانات جديدة ، فلا داعي لإعادة تشغيل الخوارزمية من البداية. بعبارات أبسط ، يحاول التعلم الاستقرائي إنشاء نموذج عام يتم فيه التنبؤ بكل نقطة بيانات جديدة بناءً على مجموعة نقاط البيانات التعليمية المرصودة. هنا يمكنك توقع أي نقطة في مساحة النقطة ، بخلاف النقاط غير المسماة. في المقابل ، يخلق التعلم المجازي نموذجًا يناسب نقاط البيانات التدريبية والتجريبية التي لوحظت بالفعل. يتنبأ هذا النهج بتسميات البيانات غير المسماة باستخدام المعرفة بالنقاط المميزة والمعلومات الإضافية. يمكن أن يكون التعلم الرمزي مكلفًا عندما يتم تقديم نقاط بيانات جديدة بواسطة دفق إدخال[1]. في كل مرة تصل فيها نقطة بيانات جديدة ، عليك إعادة تشغيل كل شيء. من ناحية أخرى ، ينشئ التعلم الاستقرائي نموذجًا تنبؤيًا مبدئيًا ، ويمكن تسمية نقاط البيانات الجديدة في وقت قصير جداً مع عدد أقل من العمليات الحسابية.

تخيل أن لديك مجموعة بيانات تدريبية ، ولكن مجموعة فرعية واحدة فقط منها مسماة. على سبيل المثال ، تحاول تصنيف ما إذا كانت هناك زهرة في الصورة. لديك 100000 صورة ، ولكن لديك فقط 1000 صورة تحتوي بالتأكيد على زهرة ، و 1000 صورة أخرى تعرف أنها لا تحتوي على زهرة ، وليس لديك فكرة عن 98000 مثال آخر. بمعنى آخر ، قد يكون لديهم أزهار أو لا. يعمل التعلم الاستقرائي عن طريق فحص 2000 عينة مميزة وبناء مصنف على 2000 عينة. يقول التعلم المجازي "انتظر ، قد لا يتم تصنيف الـ 98000 الآخرين ، لكنهم يخبرونني شيئًا عن مساحة المشكلة التي قد أكون قادرة على استخدامها للمساعدة في تحسين دقتي".

---

[1] input stream

باختصار :

- يعلم التعلم الاستقرائي النموذج بنقاط البيانات المسماة ويحاول التنبؤ بنقاط البيانات غير المسماة. في المقابل ، في التعلم المجازي باستخدام مجموعات البيانات التدريبية والتجريبية ، يتم تدريبها ويحاول التنبؤ بتسمية نقاط البيانات غير المسماة.
- في التعلم الاستقرائي ، إذا تم تقديم نقطة بيانات جديدة غير مسماة ، فيمكننا استخدام النموذج الذي تم تدريبه مسبقًا للتنبؤ. ومع ذلك ، في التعلم المجازي ، قد نحتاج إلى إعادة تدريب النموذج بأكمله.
- في التعلم المجازي ، يجب أن تكون مجموعة الاختبار متاحة مسبقًا ، بحيث تستخدم النمذجة المجازية للمشكلة المعلومات المتاحة من بيانات الاختبار غير المصنفة من أجل دقة أفضل.
- يعد التعلم المجازي أكثر تكلفة من الناحية الحسابية من التعلم الاستقرائي.

التعلم النشط[1]

التعلم النشط هو فرع من فروع التعلم الآلي حيث يمكن لخوارزمية التعلم التواصل مع المستخدم لتمييز البيانات بالمخرجات المرغوبة. يبني المتعلم الأمثلة التدريبية الخاصة به بناءً على بعض الاستراتيجيات لتحسين أدائه. في التعلم النشط ، تختار الخوارزمية باستمرار (بنشاط) مجموعة فرعية من العينات التي سيتم تصنيفها في الخطوة التالية من مجموعة البيانات غير المسماة. يتمثل الاعتقاد الرئيسي وراء فكرة خوارزمية التعلم النشط في أنها تسمح لخوارزمية التعلم الآلي بتحديد البيانات التي تحتاج إلى التعلم منها من أجل تحقيق درجة أعلى من الدقة نظريًا باستخدام عدد محدود من علامات التدريب. نتيجة لذلك ، يُسمح للمتعلمين الناجحين بطرح الأسئلة بشكل تفاعلي أثناء مرحلة التدريب. عادة ما تكون هذه الاستعلامات في شكل بيانات غير مسماة مع طلب للتعليق التوضيحي البشري لوضع العلامات. ونتيجة لذلك ، يصبح التعلم النشط جزءًا من النموذج البشري في الحلقة ، حيث يعد أحد أهم أمثلة الأداء.

الغرض من التعلم النشط هو زيادة أداء خوارزمية التعلم الآلي مع الحفاظ على عدد العينات التدريبية ثابتًا. غالبًا ما يستخدم هذا النهج عندما يكون إنتاج عينات التدريب مكلفًا أو يستغرق وقتًا طويلاً. الاستراتيجيتان الأساسيتان في التعلم النشط هما **أخذ عينات غير المؤكدة**[2] **وأخذ عينات من مساحة العينة**[3].

---

[1] Active learning

[2] uncertainty sampling

[3] version space sampling

أخذ عينات غير المؤكدة

في أخذ العينات غير المؤكدة ، يختار المتعلم بشكل متكرر عينات التدريب الخاصة به ، بحيث تكون العينات المختارة هي العينات التي يكون المتعلم أقل ثقة فيها. عادةً ما يكون لدى المتعلم نموذج يتم تدريبه على نماذجهم التعليمية الحالية التي تتنبأ باحتمالية أن تنتمي العناصر إلى فصل معين. ثم يستخدم المتعلم تقنية التصغير للعثور على نمط يقلل من هذا الاحتمال (وبالتالي فإن النموذج لديه أقل ثقة في كيفية التصنيف). تم تصنيف هذه النقطة بواسطة طرف ثالث وإضافتها إلى البرنامج التعليمي. تتكرر هذه العملية حتى يرضي المتعلم عن أداء المصنف. يعد التنفيذ والاستدلال وراء أخذ العينات من عدم اليقين أمرًا بسيطًا للغاية ، مما جعل هذه الإستراتيجية شائعة في التعلم النشط. عيب هذه الاستراتيجية هو أنه عندما يكون المتعلم متأكدًا من فئة جزء من مساحة الإدخال (بينما قد يصنف هذا الجزء بشكل غير صحيح) ، لا يختار المتعلم حالات جديدة لهذا الجزء. يمكن أن يتسبب هذا في فشل النموذج النهائي في هذا الجزء من مساحة الإدخال.

أخذ عينات من مساحة العينة

مساحة العينة للمجموعة التدريبية هي مساحة تتضمن جميع النماذج التي تتطابق مع الأمثلة التدريبية بشكل صحيح. بمعنى آخر ، مساحة العينة عبارة عن هيكل يساعد في الحفاظ على جميع الفرضيات القادرة على تصنيف ملاحظاتنا الحالية بشكل كامل. تكمن الفكرة في زيادة سرعة التعلم عن طريق اختيار العينات بطريقة تقلل مساحة العينة بسرعة مع كل علامة.

من تعريف مساحة العينة ، يستنتج أن إضافة عينة تدريب إلى مجموعة التدريب يمكن أن تقلل فقط من حجم مساحة العينة. هذا هو أساس أخذ العينات من مساحة العينة. في أخذ العينات من مساحة العينة ، يتم اختيار العينة التالية بحيث يكون تقليل حجم مساحة العينة إلى أقصى حد. يتم ذلك عن طريق اختيار العينات التي تنتمي إليها النماذج الموجودة في مساحة العينة.

التعلم الاونلاين[1]

في أبسط أشكاله ، يعد التعلم الاونلاين أسلوبًا للتعلم الآلي يتعلم أمثلة من البيانات في الوقت الفعلي من الملاحظة في وقت واحد. بمعنى آخر ، تعمل خوارزميات التعلم الاونلاين مع البيانات المتاحة. تتحسن الخوارزميات الاونلاين تدريجيًا مع وصول كل عينة بيانات جديدة ، ثم تجاهل تلك البيانات وتوقف عن استخدامها. هذا ليس اجبارياً ، ولكن من المرغوب فيه عادةً أن تنسى

[1] Online learning

الخوارزمية الاونلاين العينات القديمة بمرور الوقت حتى تتمكن من التكيف مع مجموعات العينات غير الثابتة[1].

يمكن تطبيق التعلم الاونلاين على المشكلات التي يتم فيها تقديم العينات بمرور الوقت والتنبؤ بالتوزيع الاحتمالي للعينات بمرور الوقت. يعد الانحدار الاشتقاقي للانتشار الخلفي(كما هو مستخدم في الشبكات العصبية) مثالاً على هذا النوع من التعلم.

التعلم الاوفلاين

تعمل خوارزميات التعلم الاوفلاين مع البيانات المجمعة من مجموعة البيانات. بمعنى آخر ، يجب أن يتعلم من البيانات التي لديه بالفعل. يجب إعادة تشغيل خوارزميات الاوفلاين للتعلم من البيانات المعدلة. تعد خوارزميات SVM والغابات العشوائية خوارزميات اوفلاين تمامًا (على الرغم من أن الباحثين قاموا ببناء متغيراتهم اونلاين).

الفرق بين التعلم الاونلاين والاوفلاين

يمكن تفسير الفرق بين هذين النوعين من التعلم بمثال بسيط (Dulhare et al.، 2020). افترض أن طالبًا يريد تعلم الجبر. في النوع الأول من التعلم ، يمكن لهذا الطالب قراءة وتعلم العديد من كتب الجبر. بعد التعلم من هذه الكتب ، لا يتعلم أي شيء جديد ثم يستخدم معرفته فقط. هذا نوع من التعلم الاونلاين. في طريقة التعلم هذه ، تتوفر جميع البيانات أثناء التدريب وبعد مرحلة التدريب ، لن يكون لدينا المزيد من التعلم. في المقابل ، في التعلم الثاني أو التعلم الاوفلاين، يقرأ الطالب كتبه ويقرأها أولاً. يتعلم ثم ، أثناء استخدام معرفته ، كلما وجد كتابًا جديداً في مجال الجبر ، يقرأه ، ومن خلال قراءته يحسن معدل تعلمه.

لذا فإن التعلم الاونلاين يشبه امتلاك مجموعة من الكتب وعليك أن تتعلمها. كل مصادرك هي هذه الكتب ولديك بالفعل كل البيانات ، ولكن لنفترض أنه عليك تعلم أشياء جديدة وتضيف إلى معرفتك السابقة في سياق التعايش مع المعلومات الجديدة التي تُعطى لك كل يوم. هذه هي الحالة الثانية للتعلم الاونلاين ، عندما لا تكون جميع البيانات متاحة حاليًا.

التعلم الاونلاين له فائدتان رئيسيتان:

1. **يمكن لهذه الطريقة تدريب كميات كبيرة جداً من البيانات.** على سبيل المثال ، البيانات غير الموجودة في الذاكرة بسبب الحجم الكبير.
2. **يتم تغطية التغييرات التي قد تحدث في طبيعة البيانات بهذه الطريقة.** لنفترض أن Google قد طورت خوارزمية لنظام البريد الإلكتروني الخاص بها والتي تكتشف بذكاء رسائل البريد الإلكتروني المزعجة(غير الهامة) باستخدام خوارزميات التعلم

[1] non-stationary

الآلي. كما قد تتوقع ، يتغير محتوى رسائل البريد الإلكتروني المزعجة باستمرار ، ويعمل الأشخاص الذين يرسلون رسائل البريد الإلكتروني المزعجة على تحسين أنفسهم كل يوم ضد خوارزميات Google هذه. لذلك ، يمكن لخوارزمية اكتشاف البريد الإلكتروني المزعج من Google إجراء تدريب عبر الإنترنت لاكتشاف رسائل البريد الإلكتروني المزعجة التي تغيرت بمرور الوقت. في الواقع ، يتم تحديث تعلم الخوارزمية وتحسينه عن طريق تعديل محتوى وتنسيق رسائل البريد الإلكتروني المزعجة.

من بين نوعين من الخوارزميات، تعد الخوارزميات الاونلاين هى الأكثر شيوعًا. نظرًا لأنه يمكنك بسهولة إنشاء خوارزمية اوفلاين من خوارزمية اونلاين بدقة بالإضافة إلى مجموعة بيانات مخزنة، لكن العكس ليس صحيحًا بالنسبة لخوارزمية اوفلاين تمامًا. ومع ذلك، فإن هذا لا يجعلها بالضرورة متفوقة (غالبًا ما يتم إجراء تنازلات من حيث كفاءة العينة أو تكلفة CPU أو الدقة عند استخدام خوارزمية اونلاين). يمكن اعتبار الأساليب مثل الدفعة الصغيرة فى تدريب الشبكة العصبية على أنها محاولة لإيجاد حل وسط بين الخوارزميات المتصلة الاونلاين والاوفلاين.

التعلم الجماعي[1]

في خوارزميات التعلم الجماعي ، تكون بيانات التدريب متاحة بالكامل من وكيل التعلم من البداية ، وبعد مرحلة التدريب ، لا يمكن إضافة بيانات تدريب جديدة إلى النظام. في هذه الخوارزميات ، إذا كانت بيانات التدريب كبيرة جدًا ، فستكون الدورة التدريبية طويلة وتستغرق وقتًا طويلاً وفي بعض الحالات قد لا تكون هناك مساحة كافية لتخزين بيانات التدريب بالكامل.

التعلم المتزايد[2]

في خوارزميات التعلم المتزايد ، قد لا تكون بيانات التدريب معروفة أو مكتملة من البداية أو يمكن إضافتها بمرور الوقت. بمعنى آخر ، من الممكن لهذه الخوارزميات إدخال بيانات تدريب جديدة بعد مرحلة التدريب. الغرض من هذه الخوارزميات هو الحفاظ على نتائج المراحل السابقة من التدريب وتحسين أداء عامل التعلم فقط من خلال تعلم أنماط جديدة. في الواقع ، تتكيف هذه الخوارزميات وتحدّث نفسها بإدخال خوارزميات جديدة دون الحاجة إلى إعادة تدريب الخوارزميات القديمة التي ربما لم تعد متوفرة.

---

[1] Batch Learning

[2] Incremental Learning

يعد التعلم المتزايد أحد أهم الموضوعات في التعلم الآلي. يمكن العثور على تعريفات وتفسيرات مختلفة للتعلم المتزايد في نصوص مثل التعلم الاونلاين ، أو إعادة التصنيف غير الصحيحة للأمثلة السابقة ، أو تطوير بنية المجموعات وتقليمها. الهدف العام لنظام التعلم المتزايد هو تحديث الافتراضات والمعلومات السابقة عند تقديم أمثلة جديدة ، دون إعادة استخدام الأمثلة السابقة. وتجدر الإشارة ، مع ذلك ، إلى أن مثل هذا النظام ، عند تقديم عينات جديدة ، لا ينسى نتائج التدريب لخطواته السابقة على العينات القديمة ، بل يحسن معرفته السابقة بالعينات الجديدة. بمعنى آخر ، يُطلق على التعلم المتزايد التعلم المتزايد عندما يكون له الخصائص التالية (.Dulhare et al، 2020):

- **القدرة على اكتساب معرفة إضافية عند إدخال بيانات جديدة.**
- **القدرة على الاحتفاظ بالمعلومات المستفادة من مراحل التعلم السابقة.**
- **القدرة على تعلم فئة جديدة إذا تم توفير عينة جديدة.**

بناءً على الدراسات التي أجريت على خوارزميات التعلم المتزايد، يمكن تصنيف هذه الخوارزميات وفقًا لمعايير مختلفة. على سبيل المثال ، يمكن تقسيم هذه الخوارزميات إلى ثلاث فئات بناءً على قدرتها على تخزين البيانات التعليمية.

- **البيانات الكاملة:** هذه المجموعة من الخوارزميات قادرة على تخزين جميع بيانات التدريب دون فقدان البيانات القديمة. تشمل مزايا هذه الخوارزميات ترقيات وتحديثات فعالة ودقة دقيقة. ومع ذلك ، نظرًا لتوفر جميع بيانات التدريب ، تتطلب هذه الخوارزميات مساحة تخزين كبيرة.
- **البيانات الجزئية:** تحتوي هذه الخوارزميات على بيانات معينة فقط. وبالتالي يصلون إلى درجة من التنازل بين الدقة واستخدام الذاكرة.
- **بدون بيانات:** تخزن هذه الخوارزميات المعلومات الإحصائية المتعلقة بالبيانات فقط وتتجاهل جميع البيانات. لذلك ، اعتمادًا على نوع البيانات المخزنة ، تكون دقة هذه الخوارزميات أقل من الفئتين المذكورتين أعلاه. ومع ذلك ، فإن استخدام الذاكرة لهذه الخوارزميات منخفض أيضًا.

التعلم الذاتي[1]

يمكن تسمية التعلم الذاتي بإصدار أكثر تقدمًا من التعلم غير الخاضع للإشراف الذي يتطلب بيانات تنظيمية. في هذه الحالة فقط ، لا يقوم البشر بوضع العلامات على البيانات ، وانما النموذج هو الذي يحصل على العلامات من البيانات. نظرًا لعدم وجود حاجة للتعليقات البشرية حول

[1] Self-Supervised Learning

تصنيف البيانات ، يمكن اعتبار التعلم الذاتي شكلاً مستقلاً من التعلم الخاضع للإشراف. يقوم التعلم الذاتي بوضع العلامات بمساعدة البيانات الوصفية المضمنة كبيانات تنظيمية.

التعلم المتعدد الامثلة[1]

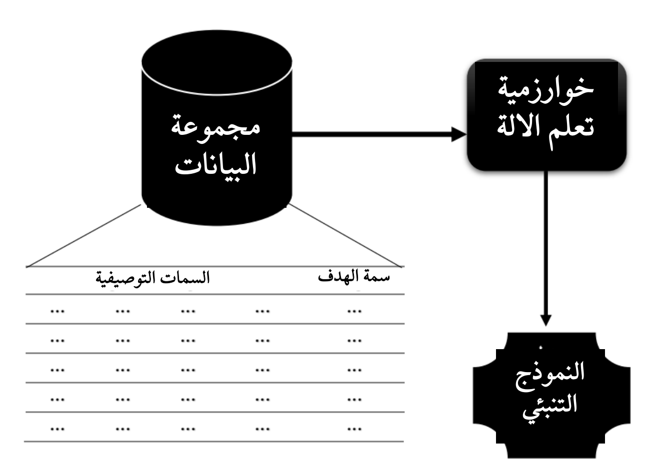
تم اقتراح التعلم متعدد الامثلة كنوع من التعلم الخاضع للإشراف للقضايا المتعلقة بالمعرفة غير الكاملة حول علامات العينات. في التعلم الخاضع للإشراف ، يتم تعيين تسمية منفصلة أو قيمة حقيقية لكل عينة تدريبية. في المقابل ، في تعلم حالات متعددة من العلامات ، يتم تخصيصها لحقائب من العينات (تسمى مجموعة منتظمة من بيانات التدريب حقيبة ، ويتم تمييز الحقيبة بأكملها). في الوضع الثنائي ، إذا كانت عينة واحدة على الأقل في تلك الحقيبة موجبة ، فإن الحقيبة تحمل ملصق موجب ، وإذا كانت جميع العينات الموجودة فيه سلبية ، فسيكون للحقيبة ملصق سلبي. بمعنى آخر ، إذا تطابقت العينة مع النتيجة ، فإن الحقيبة كلها يكون موجبًا ، وإذا لم يتطابق ، فإن الحقيبة بأكملها سلبية. الغرض من العديد من الأمثلة هو تصنيف الحقائب أو العينات غير المرئية بناءً على حقائب مصنفة على أنها بيانات تدريبية.

تعلم كيف تعمل الآلة؟

يُعرَّف التعلم الآلي بأنه عملية مؤتمتة تستخرج الأنماط من البيانات. نستخدم التعلم الآلي الخاضع للإشراف لبناء النماذج المستخدمة في برامج تحليل البيانات التنبؤية. يتعلم نهج التعلم الآلي الخاضع للإشراف تلقائيًا نموذجًا للعلاقة بين مجموعة الخصائص الوصفية والخصائص المستهدفة ، بناءً على مجموعة من الأمثلة أو الأمثلة السابقة. بعد ذلك ، يمكننا استخدام هذا النموذج للتنبؤ بأمثلة جديدة. هاتان الخطوتان منفصلتان موضحتان في الشكل 4ـ1.

يسرد الجدول 4ـ2 مجموعة من الأمثلة السابقة أو قواعد بيانات قروض الإسكان التي منحها البنك في الماضي. تحتوي مجموعة البيانات هذه على سمات وصفية تصف الرهن العقاري والسمة المستهدفة التي تشير إلى ما إذا كان مقدم طلب الرهن العقاري قد فشل في نهاية المطاف في سداد القرض أو قام بسداده بالكامل. تحتوي الخصائص الوصفية على ثلاثة سجلات (معلومات) حول الرهن العقاري: المهنة (التي يمكن أن تكون إدارية أو صناعية) ، وعمر مقدم الطلب والنسبة بين راتب مقدم الطلب والمبلغ المقترض (نسبة القرض إلى الراتب). يتم تعيين السمة الهدف (النتيجة) بشكل افتراضي أو استرداد. في مصطلحات التعلم الآلي ، يُشار إلى كل صف في مجموعة البيانات على أنه **عينة تدريب** ويشار إلى مجموعة البيانات العامة على أنها **مجموعة بيانات تدريب**.

[1] Multiple Instance Learning



(آ) تعلم نموذج من مجموعة من الأمثلة السابقة



(ب) استخدام النموذج للتوقع

الشكل 4ــ1 خطوتان في التعلم الآلي الخاضع للإشراف: (أ) التعلم و (ب) التوقع.

جدول 4ــ2 قاعدة بيانات تقييم القروض

التسلسل	المهنة	العمر	نسبة القرض إلى الراتب	النتيجة
1	صناعية	34	2.96	استرداد
2	ادارية	41	4.64	افتراضي
3	ادارية	36	3.22	افتراضي
4	ادارية	41	3.11	افتراضي
5	صناعية	48	3.80	افتراضي
6	صناعية	61	2.52	استرداد
7	ادارية	37	1.50	استرداد
8	ادارية	40	1.93	استرداد
9	صناعية	33	5.25	افتراضي
10	صناعية	32	4.15	افتراضي

فيما يلي مثال على نموذج تنبؤ بسيط للغاية لمجموعة البيانات هذه:

```
if نسبة القرض إلى الراتب > 3 then
    النتيجة = الافتراضي
else
    النتيجة = ارجاع الأموال(استرداد)
end if
```

يمكن القول أن هذا النموذج متوافق مع مجموعة البيانات هذه ، لأنه لا توجد أمثلة في مجموعة البيانات لا يتنبأ بها النموذج بشكل صحيح. عند تقديم طلبات الرهن العقاري الجديدة ، يمكننا استخدام هذا النموذج للتنبؤ بما إذا كان مقدم طلب الرهن العقاري سوف يسددها أم لا ، واتخاذ قرارات الإقراض بناءً على تلك التوقعات.

تعمل خوارزميات التعلم الآلي على أتمتة عملية تعلم النموذج ، مما يوضح العلاقة بين الخصائص الوصفية وخاصية الهدف في مجموعة البيانات. بالنسبة لمجموعات البيانات البسيطة مثل تلك المعروضة في الجدول 4ـ2 ، قد نتمكن من إنشاء نموذج تنبؤ يدويًا ، وفي مثال على هذا المقياس ، لا يعمل التعلم الآلي بالنسبة لنا. لنفترض الآن أن مجموعة البيانات نفسها تحتوي على ميزات وصفية أكثر ، مثل مبلغ الرهن العقاري الذي يتلقاها الشخص ، ورواتب المرتهن ، ونوع العقار الذي يتعلق به الرهن العقاري ، وعدد المرات التي اقترض فيها الشخص. الآن ، لم يعد نموذج التنبؤ البسيط ، الذي لم يستخدم فقط ميزة قرض يوم الدفع ، متوافقًا مع مجموعة البيانات هذه. وبالتالي ، من الصعب جدًا العثور على مثل هذه القاعدة في مجموعة البيانات هذه. بمعنى آخر ، يكاد يكون من المستحيل تعلم هذا النموذج يدويًا عن طريق فحص البيانات. هذا هو المكان الذي يلعب فيه التعلم الآلي، لأنه بسيط للغاية بالنسبة لخوارزمية التعلم الآلي. ومن ثم، عندما نريد بناء نماذج تنبؤية من مجموعات بيانات كبيرة ذات خصائص متعددة ، فإن الطريقة هي استخدام التعلم الآلي.

تبحث خوارزميات التعلم الآلي عن مجموعة من نماذج التنبؤ المحتملة في محاولة لبناء نموذج يوضح بشكل أفضل العلاقة بين الخصائص الوصفية والخصائص الموضوعية في مجموعة البيانات. المعيار الواضح لتوجيه هذا البحث هو البحث عن النماذج المتوافقة مع البيانات. ومع ذلك ، هناك سببان على الأقل لعدم فائدة البحث البسيط عن النماذج المتوافقة بما يكفي لتعلم النماذج التنبؤية. أولاً ، عند التعامل مع مجموعات البيانات الكبيرة ، من المحتمل أن يكون هناك ضوضاء في البيانات أيضًا ، والنماذج التنبؤية المتوافقة مع البيانات الصاخبة تقدم تنبؤات خاطئة. ثانيًا ، في معظم الأوقات في مشروعات التعلم الآلي ، لا تمثل مجموعة التدريب سوى عينة صغيرة من مجموعة الأمثلة المحتملة في المجال. نتيجة لذلك ، يعد التعلم الآلي **مشكلة مطروحة بشكل سيئة**[1] ، وهي مشكلة لا يمكن العثور على حل فريد لها باستخدام البيانات الموجودة وحدها. يمكننا استخدام مثال لتوضيح كيف أن التعلم الآلي هو مشكلة يريد فيها فريق التحليلات في سلسلة سوبر ماركت تصنيف عملائه إلى مجموعات عازبة أو متزوجة أو عائلية بناءً على عادات التسوق الخاصة بهم فقط. تحتوي مجموعة البيانات الواردة في الجدول 4ـ2 على ميزات وصفية تصف عادات التسوق لخمسة عملاء. توضح الميزات الوصفية في هذا الجدول ما إذا كان العميل يشتري أغذية الأطفال أو المشروبات الغازية أو المنتجات النباتية

[1] ill-posed problem

العضوية أم لا. تحتوي كل سمة على واحدة من قيمتين ، *نعم* أو *لا* . بالإضافة إلى هذه الميزات الوصفية ، هناك سمة مستهدفة تصف المجموعة الديموغرافية لكل عميل (اعزب أو زوج أو عائلة). تسمى مجموعة البيانات الواردة في الجدول 4ـ3 مجموعة بيانات معنونة لأنها تحتوي على قيم للسمة الهدف.

جدول 4ـ3 مجموعة بيانات البيع بالتجزئة البسيطة

الرقم	أغذية الأطفال	مشروب غازي	منتجات نباتية عضوية	الهدف
1	لا	لا	لا	زوج
2	نعم	نعم	نعم	عائلي
3	نعم	نعم	لا	عائلي
4	لا	لا	نعم	زوج
5	لا	نعم	نعم	اعزب

تخيل أننا نحاول تعلم نموذج تنبؤي لسيناريو البيع بالتجزئة هذا من خلال البحث عن نموذج يناسب مجموعة البيانات. أول شيء يتعين علينا القيام به هو معرفة عدد النماذج المختلفة الممكنة بالفعل لهذا السيناريو. تحدد هذه الخطوة مجموعة نماذج التنبؤ التي تبحث عنها خوارزمية تعلم الآلة. من منظور البحث عن نموذج متسق ، فإن الميزة الأكثر أهمية لنموذج التنبؤ هي أنه يحدد تعيين أي مجموعة محتملة من قيم الخصائص الوصفية في تنبؤ للخاصية الموضوعية. بالنسبة لسيناريو البيع بالتجزئة ، هناك ثلاث خصائص وصفية ثنائية فقط ، لذلك هناك $2^3 = 8$ مجموعات محتملة من قيم الخصائص الوصفية. ومع ذلك ، لكل من هذه المجموعات الثمانية الممكنة من قيم السمات الوصفية ، هناك 3 قيم سمات موضوعية محتملة. لذلك ، هذا يعني أن هناك 6566 نموذجًا ممكنًا للتنبؤ يمكن استخدامها. يوضح الجدول (أ) 4ـ3 العلاقة بين مجموعة من قيم السمات الوصفية ونماذج التنبؤ لسيناريو البيع بالتجزئة. يتم سرد مجموعة الميزات الوصفية على الجانب الأيمن من الجدول ، ويتم عرض مجموعة النماذج المحتملة لهذا المجال من M_1 إلى M_{6561} على الجانب الأيسر من الجدول. باستخدام مجموعة التدريب في الجدول 4ـ2 ، تكمل خوارزمية التعلم الآلي مجموعة 6561 من نماذج التخفيض المحتملة لهذا السيناريو فقط للنماذج التي تتوافق مع أمثلة التدريب. يوضح الجدول (ب) 4ـ3 هذا. توضح الأعمدة الفارغة في الجدول النماذج التي لا تتوافق مع بيانات التدريب. يوضح الجدول (ب) 4ـ4 أيضًا أن مجموعة بيانات التدريب لا تحتوي على عينة لكل مجموعة ممكنة من قيم الخصائص الوصفية. بمعنى آخر ، لا يزال هناك عدد كبير من نماذج التنبؤ المحتملة التي تظل متوافقة مع مجموعة بيانات التدريب بعد إزالة النماذج غير المتوافقة. أي أن هناك ثلاث مجموعات من قيم السمات الوصفية التي لا تُعرف قيمة السمة الموضوعية الدقيقة لها. لذلك ، هناك $27 = 3^3$ نموذجًا محتملاً متوافقة مع بيانات التدريب. ثلاثة من هذه الحالات M_2 ، M_4 و M_5 موضحة في الجدول. نظرًا لأنه لا يمكن العثور على نموذج واحد متماسك بناءً على عينة

مجموعة بيانات التدريب ، فإننا نقول إن التعلم الآلي هو في الأساس مشكلة مطروحة بشكل سيئة.

جدول 4ـ4 نماذج التنبؤ المحتملة (أ) قبل و (ب) بعد توافر بيانات التدريب

(آ) قبل توفر بيانات التدريب

الرقم	أغذية الأطفال	مشروب غازي	منتجات نباتية عضوية	الهدف	M_1	M_2	M_3	M_4	M_5	...	M_{6561}
1	كلا	كلا	كلا	؟	زوج	زوج	اعزب	زوج	زوج		زوج
2	كلا	كلا	نعم	؟	مجرد	زوج	اعزب	زوج	زوج		اعزب
3	كلا	نعم	كلا	؟	عائلي	عائلي	اعزب	زوج	مجرد		عائلي
4	كلا	نعم	نعم	؟	مجرد	اعزب	اعزب	زوج	مجرد		زوج
5	نعم	كلا	كلا	؟	زوج	زوج	عائلي	عائلي	عائلي		عائلي
6	نعم	كلا	نعم	؟	زوج	عائلي	عائلي	عائلي	عائلي		زوج
7	نعم	نعم	كلا	؟	اعزب	عائلي	عائلي	عائلي	عائلي		اعزب
8	نعم	نعم	نعم	؟	اعزب	مجرد	عائلي	عائلي	زوج		عائلي

(ب) بعد توفر بيانات التدريب

الرقم	أغذية الأطفال	مشروب غازي	منتجات نباتية عضوية	الهدف	M_2	M_4	M_5	...
1	كلا	كلا	كلا	زوج	زوج	زوج	زوج	
2	كلا	كلا	نعم	زوج	زوج	زوج	زوج	
3	كلا	نعم	خير	؟	عائلي	زوج	اعزب	
4	كلا	نعم	نعم	اعزب	اعزب	زوج	اعزب	
5	نعم	كلا	كلا	؟	زوج	عائلي	عائلي	
6	نعم	كلا	نعم	عائلي	عائلي	عائلي	عائلي	
7	نعم	نعم	خير	عائلي	عائلي	عائلي	عائلي	
8	نعم	نعم	نعم	؟	اعزب	عائلي	زوج	

قد نعتقد أن وجود العديد من النماذج المتوافقة مع البيانات أمر جيد. ومع ذلك ، تكمن المشكلة في أنه على الرغم من أن هذه النماذج تتفق على التنبؤات التي يجب إجراؤها لمجموعات البيانات التدريبية ، إلا أنها لا تتفق على التنبؤات التي يجب إجراؤها للحالات غير الموجودة في مجموعة البيانات التدريبية والتي يجب إرجاعها(لا تعمل بشكل جيد). على سبيل المثال ، إذا بدأ عميل جديد في التسوق في سوبر ماركت واشترى أغذية الأطفال والمشروبات الغازية والمنتجات النباتية العضوية ، فإن مجموعة النماذج الثابتة لدينا ستتعارض مع التوقعات التي يجب إعادتها إلى هذا العميل. على سبيل المثال ، تُرجع M_2 الهدف اعزباً ، وترجع M_4 الهدف عائلي المستهدفة ، وترجع M_5 الهدف زوجيًا.

لا توفر معايير التوافق مع البيانات التدريبية أي إرشادات بشأن أي من النماذج المتوافقة يتعامل مع الاستعلامات[1] التي تقع خارج مجموعة البيانات التدريبية. نتيجة لذلك ، لا يمكننا استخدام مجموعة من النماذج المتوافقة للتنبؤ بهذه الاستعلامات. في الواقع ، يعد البحث عن نماذج تنبؤية تتطابق مع مجموعة البيانات بمثابة تذكر مجموعة البيانات. وفقًا لذلك ، لا يتم التعلم بهذه الطريقة. لأن مجموعة من النماذج المتسقة لا تخبرنا شيئًا عن العلاقة الأساسية بين الخصائص الوصفية والغرض بخلاف ما توفره نظرة بسيطة على مجموعة البيانات التدريبية.

إذا كان النموذج التنبئي مفيدًا ، فيجب أن يكون قادرًا على عمل تنبؤات جيدة للاستعلامات غير الموجودة في البيانات. يُظهر النموذج التنبئي الذي يقوم بالتنبؤات الصحيحة لهذه الأسئلة العلاقة الأساسية بين السمات الوصفية والغرض ويقال إنه معمم[2] جيدًا. في الممارسة العملية ، الهدف من التعلم الآلي هو العثور على النموذج التنبئي الأكثر عمومية. للعثور على هذا النموذج الأفضل ، يجب أن تستخدم خوارزمية التعلم الآلي معايير للاختيار من بين النماذج المرشحة التي تراها أثناء البحث. بالنظر إلى أن التوافق مع مجموعة البيانات ليس معيارًا جيدًا لاختيار أفضل نموذج تنبؤي ، ما هو المعيار الذي يجب أن نستخدمه؟ هناك العديد من الإجابات المحتملة لهذا السؤال وهذا هو سبب وجود خوارزميات مختلفة للتعلم الآلي. تستخدم كل خوارزمية تعلم الآلة معايير اختيار نموذج مختلفة لتوجيه بحثها عن أفضل نموذج تنبؤي. لذلك ، عندما نختار استخدام خوارزمية تعلم الآلة بدلاً من أخرى ، في الممارسة العملية ، نختار استخدام معيار اختيار نموذج بدلاً من آخر.

تتضمن جميع معايير اختيار النموذج المختلفة مجموعة من الافتراضات حول خصائص النموذج التي نريد أن تستنتجها الخوارزمية. تُعرف مجموعة الافتراضات التي تحدد معايير اختيار النموذج لخوارزمية التعلم الآلي باسم التحيز الاستقرائي[3] لخوارزمية التعلم الآلي. هناك نوعان من التحيز الاستقرائي يمكن لخوارزمية التعلم الآلي استخدامهما: **التحيز المحدود والتحيز المفضل**.

تعريف **التحيز المحدود**

تحديد مساحة الفرضية بحيث تكون العناصر الموجودة في مساحة الفرضية محدودة ويتم تقليل عدد عمليات البحث دون التأثير على البحث.

التحيز المحدود هو القوة التمثيلية لخوارزمية أو مجموعة من الفرضيات التي ستأخذها الخوارزمية في الاعتبار.

---

1 queries

2 generalize

3 inductive bias

تعريف التحيز المفضل

تغيير طريقة البحث والبحث غير الكامل لجميع مسافات الفرضيات.

التحيز التفضيلى هو ببساطة ما تفضله خوارزمية التعلم الخاضع للإشراف. على سبيل المثال، قد تفضل خوارزمية شجرة القرار الأشجار الأقصر والأقل تعقيداً. بعبارة أخرى، هذا التحيز هو إيمان خوارزميتنا بما يصنع الفرضية الجيدة.

يحد التحيز المحدود من مجموعة النماذج التي تراها الخوارزمية أثناء عملية التعلم. يدفع التحيز المفضل خوارزمية التعلم إلى تفضيل نماذج معينة على النماذج الأخرى. النقطة المهمة هي أن استخدام التحيز الاستقرائي هو شرط أساسي لحدوث التعلم. بدون التحيز الاستقرائي ، لا يمكن لخوارزمية التعلم الآلي أن تتعلم شيئًا بخلاف ما تقدمه. باختصار ، يعمل التعلم الآلي من خلال البحث في مجموعة من النماذج المحتملة للعثور على النموذج التنبئي الذي يمتد بشكل أفضل إلى ما وراء مجموعة البيانات. تستخدم خوارزميات التعلم الآلي مصدرين للمعلومات لتوجيه هذا البحث ، مجموعة بيانات التدريب والتحيز الاستقرائي الذي تفترضه الخوارزمية.

التحيز الاستقرائى وضرورته فى التعلم

جميع طرق التعلم لها تحيز استقرائي. التحيز الاستقرائي هو عملية تعلم المبادئ العامة بناءً على أمثلة محددة. بمعنى آخر ، هذا ما تفعله أي خوارزمية للتعلم الآلي ؛ عندما يولد تنبؤًا لكل عينة اختبار غير مرئية بناءً على عدد محدود من عينات التدريب. في الواقع ، يشير التحيز الاستقرائي إلى القيود التي تفرضها الافتراضات الموضوعة في طريقة التعلم. يعني التحيز الاستقرائي أن هناك حلولًا محتملة لا يمكننا اكتشافها وبالتالي لا توجد في مساحة العينة التي ندرسها. قد يبدو هذا وكأنه قيد سيء للغاية. لكن ، في الواقع ، التحيز الاستقرائي ضروري للتعلم. للحصول على متعلم غير متحيز ، يجب أن تتضمن مساحة العينة أي فرضيات محتملة يمكن التعبير عنها. يضع هذا قيدًا شديدًا: لا يمكن أن يكون الحل الذي ينتجه المتعلم أكثر اكتمالًا من مجموعة كاملة من البيانات التدريبية. بمعنى آخر ، يمكنه تصنيف البيانات التي شاهدها بالفعل ، لكنه لن يكون قادرًا على التعميم من أجل تصنيف البيانات الجديدة والتي تم التغاضي عنها.

بدون التحيز الاستقرائى، لا يمكن للمتعلم التعميم من الأمثلة المرصودة إلى أمثلة جديدة أفضل من التخمين العشوائى.

في القرن الرابع عشر ، اقترح ويليام إوكام "**شفرة أوكام**[1]" ، والتي تنص ببساطة على أنه من الأفضل اختيار أبسط فرضية لشرح أي ظاهرة. يمكننا اعتبار هذا نوعًا من التحيز الاستقرائي الذي

[1] Occam's Razor

ينص على أن أفضل فرضية لتناسب مجموعة من البيانات التدريبة هي أبسط فرضية. هذا يعني أنه إذا كان للخوارزميتين نفس الأداء تقريبًا لمعايير التقييم في مشروع معين ، فيجب أن نفضل الخيار "الأبسط".

ولكن ماذا تعني كلمة "أبسط" في هذا السياق؟ بشكل عام ، من الأسهل تفسير خوارزمية لديها أقل تعقيد للاستدلال (لأنها تستخدم متغيرات أقل ، على سبيل المثال ، وتتطلب هندسة ميزات أقل) ويسهل تفسيرها. ومع ذلك ، تجدر الإشارة إلى أن هذه الأنواع من التبادلات عادة ما تكون منطقية فقط عندما تكون دقة النموذج أبسط ، على الأقل ضمن نفس النطاق مثل النموذج الأكثر تعقيدًا. في سيناريو حيث قد ينتج عن نصف الدقة ملايين الدولارات في الإيرادات الإضافية أو التوفير في التكاليف ، يمكنك اختيار نموذج يصعب تفسيره أو يتطلب مزيدًا من الوقت لتطويره.

في أي مشروع للتعلم الآلي ، من الأفضل دائمًا التركيز أولاً على مشكلة العمل التي تنوي معالجتها والبدء بصياغة مقاييس النجاح الرئيسية للتحليل. بافتراض أن جميع المقاييس الرئيسية الأخرى متساوية (تقريبًا) ، استخدم شفرة اوكام وحدد النموذج الأسهل في التفسير والشرح والتطوير والصيانة. بعبارة أخرى ، تفضل النموذج الأبسط الذي يكون دقيقًا بدرجة كافية ، ولكن تأكد من أنك تعرف مساحة المشكلة جيداً بما يكفي لتعرف ما تعنيه عبارة "دقيقة بدرجة كافية" في الممارسة العملية. لأنه ، كما قال آينشتاين (ربما أعظم تلميذ أوكام) ذات مرة ، "يجب أن يكون كل شيء بسيطًا قدر الإمكان ، ولكن ليس بهذه البساطة.".

تطبيقات تعلم الآلة

يعد تعلم الآلة كلمة شائعة في تكنولوجيا اليوم وهو يتزايد يومًا بعد يوم. في الصناعة ، مهد ا تعلم الآلة الطريق للتقدم التكنولوجي والأدوات التي كانت مستحيلة قبل بضع سنوات. بالإضافة إلى ذلك ، نستخدم تعلم الآلة في حياتنا اليومية دون أن ندرك ذلك. فيما يلي قائمة ببعض أشهر تطبيقات تعلم الآلة في العالم الحقيقي.

التعرف على الصور

يعد التعرف على الصور أحد أكثر تطبيقات التعلم الآلي شيوعًا وأهمية. يتم استخدامه لتحديد الأشياء والأشخاص والأماكن وما إلى ذلك. تستخدم هذه التقنية لمزيد من التحليل مثل التعرف على الأنماط أو التعرف على الوجوه.

التعرف على الكلام

التعرف على الكلام هو عملية تحويل التعليمات الصوتية إلى نص ، يُعرف أيضًا باسم "الكلام إلى نص" أو "التعرف على كلام الكمبيوتر". حاليًا ، تُستخدم خوارزميات التعلم الآلي على نطاق واسع في العديد من برامج التعرف على الكلام..

اقتراح المنتجات

من أشهر تطبيقات التعلم الآلي التوصية بالمنتج. تعتبر توصية المنتج ميزة بارزة في كل موقع ويب للتجارة الإلكترونية تقريبًا اليوم ، وهو تطبيق متقدم لتقنيات التعلم الآلي. باستخدام التعلم الآلي والذكاء الاصطناعي ، تتعقب مواقع الويب سلوكك بناءً على عملية الشراء السابقة ونمط البحث وسجل عربة التسوق وتقديم توصيات المنتج.

ترجمة اللغة

يلعب التعلم الآلي دورًا مهمًا في ترجمة لغة إلى أخرى. اليوم ، إذا قمنا بزيارة مكان جديد ولا نعرف اللغة ، فهذه ليست مشكلة على الإطلاق ، لأن التعلم الآلي يساعدنا من خلال ترجمة النص إلى لغات معروفة. يقدم GNMT[1] من Google هذه الميزة وهي التعلم الآلي العصبي.

الخدمات المالية

تستخدم البنوك والشركات المالية الأخرى تقنية التعلم الآلي لغرضين رئيسيين: تحديد رؤى قيمة في البيانات وتقليل المخاطر. يمكن أن تحدد هذه الأفكار فرص الاستثمار أو تساعد المستثمرين في العثور على الوقت المناسب للتداول. بالإضافة إلى ذلك ، يمكنه تحديد العملاء المعرضين لمخاطر عالية أو استخدام التحليلات الإلكترونية لاكتشاف تنبيهات الاحتيال.

النفط والغاز

لا يزال البحث عن مصادر جديدة للطاقة ، وتحليل المعادن على الأرض ، والتنبؤ بفشل مستشعرات التكرير ، وتسهيل توزيع النفط لزيادة الإنتاجية والتكلفة ، والعديد من التطبيقات الأخرى في التعلم الآلي آخذة في التوسع.

المواصلات

يلعب تحليل البيانات دورًا مهمًا في تحديد الاتجاهات والأنماط في صناعة النقل ويساعد على تبسيط الطرق وتوقع مشكلات الربحية المحتملة. بالنسبة لشركات الشحن والنقل العام ومؤسسات النقل الأخرى ، يعد تحليل البيانات ونمذجة التعلم الآلي أدوات مهمة.

الرعاية الصحية

التنافس اليوم هي استخدام التعلم الآلي للتحليل الطبي. تبحث العديد من الشركات الناشئة المختلفة عن فوائد التعلم الآلي للبيانات الضخمة لتوفير الرعاية الطبية الأكثر احترافًا بهدف

[1] Google Neural Machine Translation

مشترك وهو اتخاذ القرارات الأكثر منطقية. اليوم ، يمكن لعدد لا يحصى من المستهلكين ، حتى مع هواتفهم الذكية ، قياس مجموعة واسعة من المعلومات الصحية على أساس منتظم. يمكن أن توفر أنظمة التعلم الآلي نموذجًا للحالة الصحية للشخص واستخدام التوصيات التي يقوم النظام بتحديثها لتحسين صحة الشخص.

العلاقة بالمجالات الأخرى

يعد التعلم الآلي مجالًا متعدد التخصصات وفي الواقع هناك ارتباط وثيق بين هذا المجال والعلوم الأخرى ، على سبيل المثال ، تتم مناقشة بعض هذه المجالات المهمة المتعلقة بالتعلم الآلي في استمرار هذا القسم.

الذكاء الاصطناعي

الذكاء الاصطناعي هو في الأساس نظام يبدو ذكيًا. ومع ذلك ، فإن هذا التعريف ليس دقيقًا وجيدًا للغاية. لكن ماذا يعني الذكاء الاصطناعي بالضبط؟ وفقًا لقاموس كولينز ، فإن الذكاء الاصطناعي هو "محاكاة البشر وسلوكياتهم العقلية بواسطة برنامج كمبيوتر." بعبارات أبسط ، نظام يمكنه محاكاة السلوك البشري. تشمل هذه السلوكيات حل المشكلات والتعلم والتخطيط ، والتي يتم تحقيقها ، على سبيل المثال ، من خلال تحليل البيانات وتحديد الأنماط داخلها من أجل تكرار تلك السلوكيات.

بمعنى آخر ، الكود أو التقنية أو الخوارزمية التي يمكن أن تحاكي فئة الفهم المعرفي التي تظهر في حد ذاتها أو في إنجازاته هي الذكاء الاصطناعي. ومن ثم ، فإن الذكاء الاصطناعي هو لبنة بناء التعلم الآلي. في الواقع ، يعد التعلم الآلي مجموعة فرعية رئيسية من الذكاء الاصطناعي ويمكنه تمكين الآلات من استخدام الأساليب الإحصائية لجعل تجاربهم أكثر جودة ودقة. يسمح هذا لأجهزة الكمبيوتر والآلات بتنفيذ الأوامر بناءً على بياناتهم وتعلمهم. تم تصميم هذه البرامج أو الخوارزميات لمعرفة المزيد بمرور الوقت والتكيف والتكيف مع البيانات الجديدة.

التنقيب فی البیانات [1]

يستخدم التعلم الآلي والتنقيب في البيانات نفس تقنيات التداخل. ومع ذلك ، بينما يركز التعلم الآلي القائم على الميزات على التعلم من البيانات التدريبية ، يركز التنقيب في البيانات على العثور على ميزات البيانات غير المعروفة (هذه هي خطوة لاستخراج المعرفة في قاعدة البيانات). يستخدم التنقيب في البيانات العديد من أساليب التعلم الآلي ولكن لأغراض مختلفة. من ناحية

[1] data mining

أخرى ، يستخدم التعلم الآلي أيضًا تقنيات التنقيب عن البيانات كتعلم غير خاضع للإشراف أو كخطوة معالجة مسبقة لتحسين دقة المتعلم.

ما هو الشيء المشترك بين التعلم الآلي والتنقيب في البيانات؟

يتم استخدام كل من التنقيب في البيانات والتعلم الآلي في علم البيانات ، وهو أمر منطقي لأن كلاهما يستخدم البيانات. إنهما يكملان بعضهما البعض ، وكلاهما لديه الكثير من القواسم المشتركة ، لكنهما يحققان أهدافًا مختلفة. يحظى كل من التنقيب في البيانات والتعلم الآلي بشعبية في التسويق ، واكتشاف بطاقات الائتمان المزيفة ، والتجارة الإلكترونية ، وتجارة التجزئة. يستخدم علماء البيانات ومهندسو البيانات كليهما لمساعدة الشركات. على سبيل المثال ، كل من التعلم الآلي واستخراج البيانات يجعل إدارة المخزون ومراقبة الجودة والكفاءة التشغيلية ممكنة دون تدخل بشري. عندما يتعلق الأمر بالتعلم الآلي واستخراج البيانات ، فهناك الكثير من التداخل ويستخدمه الناس (عن طريق الخطأ) بدلاً من بعضهم البعض. لكن من المهم فهم الاختلافات بينهما ، لأنك ستستخدم عمليات وبنى مختلفة اعتمادًا على أهدافك ومواردك. عندما تستخدم التعلم الآلي واستخراج البيانات بشكل صحيح ، فأنت على الطريق الصحيح لتحويل البيانات الأولية إلى رؤى قيمة ستؤثر على نتائجك. يمكن أن تكون هذه الرؤى تشغيلية أو استراتيجية أو إحصائية. على سبيل المثال ، في المستودع ، نستخدم التنقيب في البيانات والتعرف على الأنماط لحل مشكلات التوجيه المنتقاة[1]. في هذا السيناريو ، يستخدم التنقيب في البيانات تقنيات التعلم الآلي لتقدير طول أقصر مسار ممكن بدقة لزيادة الكفاءة.

التنقيب في البيانات هو قلب الذكاء الاصطناعي والتعلم الآلي والتعلم العميق والإحصاءات. بينما اشتهرت في الثلاثين عامًا الماضية ، لها تاريخ لأكثر من 200 عام. يستخدم علماء البيانات تقنيات استخراج البيانات للعثور على أنماط مخفية ولكنها مفيدة في قواعد البيانات الكبيرة التي لا يمكننا معالجتها من خلال تقنيات الاستعلام وإعداد التقارير. نظرًا لأن البيانات تنمو بشكل سريع وأسي ، يجب علينا استخدام هذه الأساليب للتحليل والتنبؤ بشكل مفيد. تساعد تقنيات التعلم الآلي في معالجة البيانات بسرعة والحصول على نتائج أسرع بشكل تلقائي. تسلط تقنيات التنقيب في البيانات الضوء على الأنماط والاتجاهات في مجموعات البيانات السابقة للتنبؤ بالنتائج المستقبلية. هذه النتائج في شكل مخططات ورسوم بيانية والمزيد.

ما هو الفرق بين التعلم الآلي والتنقيب في البيانات؟

يعد التعلم الآلي جزءًا من الذكاء الاصطناعي الذي يمنح الأنظمة القدرة على التعلم والتحسين تلقائيًا بناءً على الخبرة. في هذا السيناريو ، يمكننا بناء خوارزميات معقدة تعالج مجموعات

---

[1] picker routing problems

البيانات الكبيرة وتستخدمها للتعلم بدون برمجة واضحة. يستخدم التعلم الآلي خوارزميات معقدة يتم تعلمها والتنبؤ بها من خلال التجربة. تتحسن هذه الخوارزميات الذكية باستمرار من خلال إدخال بيانات تدريبية. الغرض الرئيسي من التنقيب هو فهم البيانات وبناء النماذج التي تتعلم العلاقات بين نقاط البيانات. في المقابل ، يصف التنقيب في البيانات ، أو عملية اكتشاف المعرفة، فعل الاستكشاف بين مجموعات البيانات. هذا النهج هو أسلوب شائع لاكتشاف الأنماط والاتجاهات غير المعروفة. يتجاوز اكتشاف المعرفة هذا في قواعد البيانات مجرد تحليل بسيط. هذا يعني أن التنقيب في البيانات يستخرج البيانات القابلة للاستخدام من مجموعة أوسع من البيانات الخام.

يتمثل الاختلاف الرئيسي بين التعلم الآلي والتنقيب في البيانات في مستوى التدخل البشري المطلوب لإكمال المهمة. التعلم الآلي ، القائم على الذكاء الاصطناعي ، هو جهاز كمبيوتر يحل محل البشر للقيام بشيء ما. في المقابل ، يتطلب التنقيب في البيانات تدخلًا بشريًا لإكمال المهمة. في هذا السيناريو ، يستخدم علماء البيانات أدوات لاستخراج واكتشاف الأنماط المفيدة في البيانات. في هذه الحالة ، هناك مجال كبير للخطأ البشري. بالمقارنة ، النتائج التي يتم الحصول عليها من خلال التعلم الآلي أكثر دقة من التنقيب في البيانات. يستخدم التعلم الآلي النماذج التنبؤية والخوارزميات الإحصائية والشبكات العصبية لتحقيق ذلك. يستخدم التنقيب في البيانات مستودعات البيانات وتقنيات تقييم الأنماط للعثور على رؤى قيمة. يمكن العثور على الاختلافات الرئيسية بين التعلم الآلي واستخراج البيانات في التطبيق والمفاهيم والتنفيذ وقدرات التعلم:

- **التطبيق:** تتطلب خوارزميات التعلم الآلي بيانات في شكل بيانات عادية (قياسية). لتحليل البيانات عن طريق التعلم الآلي ، تحتاج إلى تحويل مجموعة البيانات من شكلها الأصلي إلى تنسيق قياسي. هذا يساعد الخوارزميات الذكية على فهم البيانات بسرعة. يتطلب التعلم الآلي أيضًا كميات هائلة من البيانات لتقديم نتائج دقيقة. يمكن أيضًا أن ينتج عن التنقيب في البيانات نتائج ، ولكن بكميات أصغر من البيانات.
- **المفاهيم:** تعتمد خوارزميات التعلم الآلي على مفهوم أن الآلات تتعلم من البيانات الموجودة. يساعد هذا النهج أيضًا على تحسين الذات. يطور التعلم الآلي النماذج بناءً على المنطق الكامن وراء البيانات. هذا يساعد على التنبؤ بالنتائج المستقبلية (باستخدام طرق التنقيب في البيانات). في المقابل ، يركز التنقيب في البيانات على استخراج المعلومات باستخدام تقنيات تساعد في تحديد الأنماط والاتجاهات في البيانات.
- **التنفيذ:** يمكننا تنفيذ التعلم الآلي باستخدام الخوارزميات الذكية مثل الانحدار الخطي ، وشجرة القرار ، والشبكات العصبية ، إلخ. يستخدم التعلم الآلي أساسًا الخوارزميات الآلية والشبكات العصبية للتنبؤ بالنتائج. في المقابل ، عندما يتعلق الأمر بالتقيب في البيانات ،

نحتاج إلى بناء نماذج باستخدام قواعد البيانات ومحركات تنقيب في البيانات[1] وتقنيات تقييم الأنماط.

- **القدرة على التعلم:** يستخدم التعلم الآلي تقنيات مماثلة للتنقيب في البيانات ، ولكن الطريقة الأولى مؤتمتة. هذا يعني أن التعلم الآلي يتعلم تلقائيًا ويتكيف ويتغير. نتيجة لذلك ، يكون أكثر دقة من التنقيب في البيانات عند التنبؤ. في المقابل ، يتطلب التنقيب في البيانات تحليلًا بشريًا ويحولها إلى طريقة يدوية.
- **العامل البشري:** هناك فرق كبير نسبيًا هنا. يعتمد التنقيب في البيانات على التدخل البشري ويتم إنشاؤه في النهاية للاستخدام من قبل الأفراد. ومع ذلك ، فإن السبب الرئيسي للتعلم الآلي هو أنه يمكنه تدريب نفسه ولا يعتمد على التأثير البشري أو الفعل (التواصل البشري مع التعلم الآلي يقتصر تقريبًا على إعداد الخوارزميات الأساسية).

الإحصاء

يرتبط الإحصاء والتعلم الآلي ارتباطًا وثيقًا. ومع ذلك ، فإن الأهداف التي يسعون جاهدين لتحقيقها مختلفة تمامًا. الغرض من الإحصاء هو استنتاج عينة من المجتمع. في المقابل ، يتم استخدام التعلم الآلي للتنبؤات المتكررة من خلال إيجاد أنماط في البيانات. بالإضافة إلى ذلك ، يتطلب التعلم الآلي كميات كبيرة من البيانات لتكون قادرة على عمل تنبؤات دقيقة. يتم إنشاء النماذج باستخدام بيانات التدريب ، ويتم ضبطها بدقة[2] باستخدام مجموعة بيانات التحقق من الصحة ، وتقييمها باستخدام مجموعة بيانات تجريبية. كل هذه الخطوات تساعد الآلة على التعلم. في المقابل ، لا تتضمن الإحصائيات مجموعات فرعية متعددة ، لأنها لا تحاول التنبؤ. الغرض من النمذجة في هذه الحالة هو إظهار العلاقة بين البيانات ومتغير النتيجة.

أدوات وبيئة العمل[3] للتعلم الآلي

بيئة التعلم الآلي هي واجهة أو مكتبة أو أداة تسمح للمطورين والعلماء ببناء ونشر نماذج التعلم الآلي بسهولة وسرعة أكبر؛ دون الخوض في خوارزميات البنية التحتية. إنها توفر طريقة واضحة وموجزة لتحديد نماذج التعلم الآلي باستخدام مجموعة من المكونات الجاهزة والمحسنة. بعض الميزات الرئيسية لبيئة تعلم الآلة الجيدة هي:

---

[1] data mining engines

محرك التنقيبفي البيانات هو القلب الحقيقي لهندسة استخراج البيانات. يتضمن ذلك الأدوات والبرامج المستخدمة لاكتساب البصيرة والمعرفة من البيانات التي تم الحصول عليها من مصادر البيانات والمخزنةفي مستودعات البيانات.

[2] fine tuned

[3] Framework

- للأداء الأمثل.
- إنه صديق للمطورين.
- يسهل فهمها وترميزها.
- توفير التوازي لتوزيع العملية الحسابية.

بشكل عام، تقلل بيئة العمل الفعالة للتعلم الآلي من تعقيد التعلم الآلي وتجعله أكثر سهولة للمطورين.

على مدى السنوات القليلة الماضية، شهدنا انفجارًافي أدوات التطوير والأنظمة الأساسية المتعلقة بالتعلم الآلي والذكاء الاصطناعي. من المكتبات إلى بيئات العمل والنماذج المدربة مسبقًا. لذلك، لدى المطورين العديد من الخيارات لإدخال الذكاء الاصطناعيفي تطبيقاتهم. الغرض من هذا القسم هو تقديم بيئات تعلم الآلة المختلفة للمطورين مع التركيز على ميزاتهم الفريدة.

كيفية اختيار بيئة العمل المناسبة للتعلم الآلي

ظهرت العديد من بيئات عمل التعلم الآلي لتبسيط تطوير ونشر تطبيقات التعلم الآلي. ومع ذلك، يتعين على المطورين اتخاذ خيارات صعبةفي اختيار بيئة العمل المناسبة. لأنه قد يرغب البعض في التركيز على سهولة الاستخدام عند تدريس خوارزمية التعلم الآلي، بينما قد يعطي البعض الآخر الأولوية لتحسين المعاملات الفائقة ونشر الإنتاج. لذلك، هناك العديد من النقاط الأساسية التي يجب مراعاتها عند اختيار بيئة عمل التعلم الآلي لمشروعك:

- **تقييم احتياجاتك.** اطرح هذه الأسئلة الثلاثة عند بدء البحث عن أفضل بيئة للتعلم الآلي:

1. هل مساحة العمل هذه مستخدمة للتعلم العميق أو خوارزميات التعلم الآلي التقليدية؟
2. ما هي لغة البرمجة المفضلة لتطوير نماذج التعلم الآلي؟
3. ما هي الأجهزة والبرامج والخدمات السحابية المستخدمة في القياس؟

بايثون و R هي لغات مستخدمة على نطاق واسعفي التعلم الآلي، ومع ذلك، تتوفر أيضًا لغات أخرى مثل Julia و C و Java و Scala. تتم كتابة معظم تطبيقات التعلم الآلي اليوم بلغة بايثون ويتم نقلها من R. لأن R مصمم من قبل الإحصائيين والعمل معها غير مريح إلى حد ما. بايثون هي لغة برمجة أكثر حداثة وبسيطة وموجزة وأسهلفي الاستخدام.

- **تحسين المعاملات الفائقة.** هناك اعتبار رئيسي آخر عند اختيار بيئة عمل التعلم الآلي وهو تحسين المعلمات الفائقة. كل خوارزمية تعلم الآلة لها نهج مختلف لتحليل البيانات التدريبية وتطبيق ما تتعلمه على أمثلة جديدة. تحتوي الخوارزميات على معاملات فائقة يمكنك التفكير فيها كلوحة معلومات بها مفاتيح واتصال يتحكم في كيفية عمل الخوارزمية. يقوم

بضبط وزن المتغيرات التي يجب أخذها في الاعتبار ، وتحديد مقدار ما يجب مراعاته ، وإجراء تعديلات أخرى على الخوارزمية. عند اختيار بيئة تعلم الآلة ، من المهم مراعاة ما إذا كان هذا الإعداد يجب أن يكون تلقائيًا أم يدويًا.

- **التوسع في التطوير والتدريب.** في مرحلة تدريب الخوارزمية هناك قابلية التوسع ، وكمية البيانات التي يمكن تحليلها ، وسرعة التحليل. يمكن تحسين الأداء من خلال الخوارزميات الموزعة والمعالجة من خلال استخدام وحدات معالجة الرسوميات. في مرحلة تطوير مشروع التعلم الآلي ، ترتبط قابلية التوسع بعدد المستخدمين أو التطبيقات التي يمكنها الوصول إلى النموذج في وقت واحد في وقت واحد. نظرًا لوجود متطلبات مختلفة في مرحلة التدريب والتطوير ، تميل المؤسسات إلى تطوير نماذج في نوع واحد من البيئة (على سبيل المثال ، مساحة عمل التعلم الآلي المستندة إلى بايثون والتي يتم تشغيلها في السحابة). لذلك ، عند اختيار بيئة العمل ، من المهم النظر فيما إذا كانت تدعم كلا النوعين من قابلية التوسع.

أدوات التعلم الآلي الأكثر شيوعًا

في هذا القسم ، نلقي نظرة على بعض أدوات التعلم الآلي الأكثر شيوعًا المستخدمة اليوم.

SciKit-Learn

Scikit-learn هي واحدة من أقدم بيئات التعلم الآلي التي طورها David Cornapo كمشروع Google Code الصيفي في عام 2007. متاح كمكتبة بايثون ، وهو يدعم خوارزميات التعلم الخاضعة للإشراف وغير الخاضعة للإشراف. Scikit-Learn هي أفضل بيئة عمل لمطوري بايثون لتعلم أساسيات التعلم الآلي. تسهل مجموعة الأدوات هذه تنفيذ الخوارزميات الشائعة



مثل الانحدار الخطي ، والانحدار اللوجستي ، وأقرب جار ، وآلة متجه الدعم ، والغابة العشوائية، وشجرة القرار. بصرف النظر عن التعلم الخاضع للإشراف ، يمكن استخدام Scikit-Learn للتعلم غير الخاضع للإشراف ويدعم الخوارزميات مثل التجميع وتحليل المكونات الرئيسية والمزيد. نظرًا لأن Scikit-Learn يتعامل فقط مع تقنيات التعلم الآلي التقليدية التي لا تستخدم التعلم العميق للتدريب ، فإنه لا يتطلب وحدة معالجة الرسومات. يمكن لمطوري بايثون البدء بسرعة مع Scikit-Learn عن طريق تثبيت الحزمة. حتى المطورين الذين يستخدمون TensorFlow أو Keras أو PyTorch للتدريب يفضلون Scikit-Learn للدوال الإضافية مثل المعالجة المسبقة للبيانات والتشفير والتحقق المتبادل وتعديل المعاملات الفائقة.

المميزات

- يوفر نماذج وخوارزميات للتصنيف ، والانحدار ، والتجميع ، وتقليل الأبعاد ، واختيار النموذج ، والمعالجة المسبقة.
- يساعد في التنقيب في البيانات وتحليل البيانات.

PyTorch

PyTorch هي بيئة تعلم آلي تعتمد على Torch وهي مثالية لتصميم الشبكة العصبية. تم تطوير PyTorch بواسطة مختبر أبحاث الذكاء الاصطناعي في Facebook ، وتم إصداره في يناير 2016 كمكتبة مجانية مفتوحة المصدر ، ويستخدم بشكل أساسي في رؤية الكمبيوتر والتعلم العميق وتطبيقات معالجة اللغة الطبيعية، ويدعم تطوير البرامج المستندة إلى السحابة. يعد تنفيذ شبكة عصبية في PyTorch أبسط وأكثر سهولة من البيئات الأخرى. مع دعم وحدة المعالجة المركزية ووحدة معالجة الرسوميات، يمكن تدريب الشبكات العصبية العميقة المعقدة بمجموعات بيانات كبيرة.



المميزات

- يوفر مجموعة متنوعة من خوارزميات التحسين لبناء الشبكات العصبية.
- يمكن استخدام PyTorch على المنصات السحابية.

TensorFlow

TensorFlow هي واحدة من أشهر بيئات التعلم الآلي والتعلم العميق التي يستخدمها المطورون والباحثون. تم إطلاق TensorFlow لأول مرة بواسطة فريق Google Brain في عام 2007 ويمكن تشغيله على وحدات المعالجة المركزية ومسرعات الذكاء الاصطناعي المتخصصة ، بما في ذلك وحدات معالجة الرسومات (GPU) و (TPU). يتوفر TensorFlow على أنظمة Linux و macOS و Windows 64 بت ومنصات الحوسبة المحمولة ، بما في ذلك Android و iOS. يمكن نشر النماذج المدربة على TensorFlow على أجهزة سطح المكتب والمتصفحات وحتى وحدات التحكم الدقيقة. هذا الدعم الشامل يجعل TensorFlow فريداً وجاهزًا للانطلاق. سواء كنت تعمل على رؤية الكمبيوتر أو معالجة اللغة الطبيعية أو نماذج السلاسل الزمنية ، فإن TensorFlow عبارة عن منصة تعلم آلي قوية وعالية الأداء.



المميزات

- **النشر على منصات متعددة.** يمكن تثبيت TensorFlow على أجهزة سطح المكتب والمتصفحات وحتى وحدات التحكم الدقيقة.
- **التدريب الموزع.** يوفر TensorFlow دعمًا قويًا للتدريب الموزع في وحدة المعالجة المركزية ووحدة معالجة الرسومات.
- **تدريب شبكة عصبية موازية.** يوفر TensorFlow خطوط أنابيب تتيح لك تدريب شبكات عصبية متعددة ووحدات معالجة رسومات متعددة بشكل متوازٍ.

العيوب

- من الصعب التعلم.
- قد يكون فهم بعض رسائل الخطأ في TensorFlow أمرًا صعبًا للغاية.

Keras

Keras هي واجهة برمجة تمكن علماء البيانات من الوصول بسهولة إلى منصة التعلم الآلي TensorFlow واستخدامها. إنها واجهة برمجة تطبيقات (API) ومساحة عمل تعلم عميق مفتوحة المصدر مكتوبة بلغة بايثون تعمل على TensorFlow وتم دمجها الآن في النظام الأساسي. دعمت Keras سابقًا العديد من backend ولكنها ارتبطت حصريًا بـ TensorFlow منذ إصدار الإصدار 2.4.0 في يونيو 2020. تم تصميم Keras ، باعتبارها واجهة برمجة تطبيقات عالية المستوى ، لإجراء تجارب سريعة وسهلة تتطلب برمجة أقل من خيارات التعلم العميق الأخرى. الهدف هو تسريع تنفيذ نماذج التعلم الآلي ، ولا سيما الشبكات العصبية العميقة ، من خلال عملية تطوير "معدل التكرار العالي". يمكن تشغيل طرازات Keras على وحدة المعالجة المركزية أو وحدة معالجة الرسوميات ونشرها على منصات متعددة ، بما في ذلك متصفحات الويب وأجهزة Android و iOS المحمولة. تعد Keras أبطأ من TensorFlow و PyTorch ولكنها تتميز ببنية بسيطة وهي أكثر قابلية للقراءة وموجزة وسهلة الاستخدام وقابلة للتطوير. يعد Keras أكثر ملاءمة لمجموعات البيانات الصغيرة ويوصى به للمبتدئين نظرًا لتصميمه البسيط والمفهوم.



المميزات

- تركز على تجربة المستخدم.
- سهولة إنتاج النماذج.
- دعم الشبكات الالتفافية.
- دعم الشبكات المتكررة.
- مساحة عمل قائمة على لغة بايثون تسهل التصحيح والاستكشاف.

- تم تطويره مع التركيز على القدرة على الاختبار بسرعة.
- يمكن استخدامه لنماذج أولية سهلة وسريعة.
- يدعم الجمع بين شبكتين.
- سهل الاستخدام وقابل للتطوير.

Colab

Colaboratory ، أو Colab ، عبارة عن منتج بحثي من Google (خدمة سحابية) يسمح للمطورين بكتابة وتنفيذ كود بايثون من خلال متصفحهم. يعد Google Colab أداة رائعة لمهام التعلم العميق ويساعد على تطوير النماذج باستخدام مكتبات متعددة مثل Keras و Pytorch و OpenCv و Tensorflow والمزيد. Colab عبارة عن دفتر ملاحظات يستند إلى Jupyter ولا يحتاج إلى تثبيت ولديه إصدار مجاني رائع يوفر وصولاً مجانيًا إلى موارد حوسبة Google مثل GPU و TPU.



لماذا يجب ان نستخدم Colab؟

يعتبر Colab مثاليًا لكل شيء بدءًا من تحسين مهارات برمجة بايثون إلى العمل مع مكتبات التعلم العميق ، مثل PyTorch و Keras و TensorFlow و OpenCV. يمكنك إنشاء وتحميل وحفظ ومشاركة دفاتر الملاحظات في Colab ، وتثبيت Google Drive الخاص بك واستخدام كل ما تحفظه هناك ، وتحميل دفاتر الملاحظات مباشرة من GitHub ، وتحميل ملفات Kaggle ، وتحميل دفاتر الملاحظات الخاصة بك.

ميزة أخرى رائعة في Google Colab هي ميزة التعاون[1]. إذا كنت تعمل مع مطورين متعددين في مشروع ما ، فإن استخدام دفاتر Google Colab يعد أمرًا رائعًا. تمامًا مثل التعاون في مستند Google Docs، يمكنك البرمجة مع مبرمجين اخرين باستخدام دفتر ملاحظات Colab. بالإضافة إلى ذلك ، يمكنك أيضًا مشاركة عملك المكتمل مع مطورين آخرين.

باختصار ، يمكن سرد الأسباب المختلفة لاستخدام Colab على النحو التالي:

- مكتبات مثبتة مسبقًا.
- مخزنة في السحابة.
- التعاون.
- استخدام GPU و TPU مجانًا.

---

[1] collaboration

ومع ذلك ، هناك سيناريوهان يجب عليك استخدام Jupyter Notebook في جهازك:

1. تجنب Google Colab إذا كنت تهتم بالخصوصية وترغب في الحفاظ على سرية رمزك.
2. إذا كان لديك جهاز قوي بشكل لا يصدق مع توفر GPU و TPU.

تشغيل Google Colab

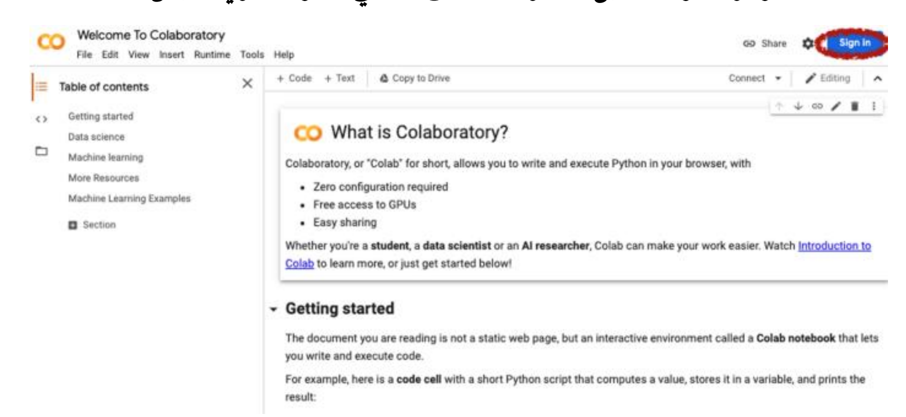
تعد عملية إعداد Colab سهلة نسبيًا ويمكن إكمالها بالخطوات التالية على أي نوع من الأجهزة:

1. **قم بزيارة صفحة Google Colab:**

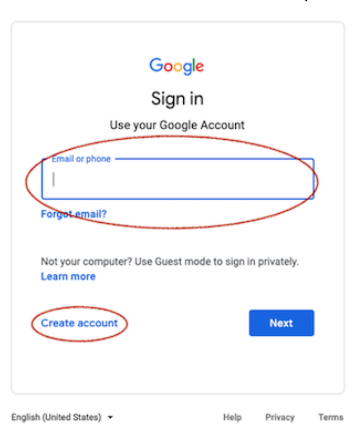
http://colab.research.google.com

سينقلك تحميل الموقع أعلاه إلى صفحة الترحيب الخاصة بـ Google Colaboratory.

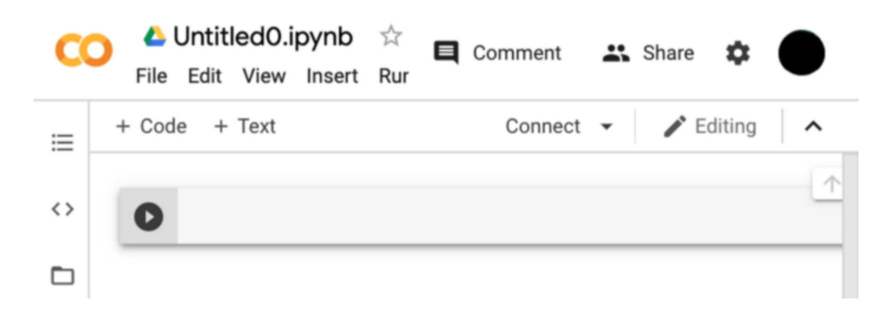
2. **انقر فوق الزر تسجيل الدخول (Sign in) في الجزء العلوي الأيمن:**



3. **قم بتسجيل الدخول باستخدام حساب GMail الخاص بك. إذا لم يكن لديك حساب GMail ، فقم بإنشاء حساب:**



4. **بمجرد الانتهاء من تسجيل الدخول ، تكون جاهزًا لاستخدام Google Colab.**
5. **من خلال النقر فوق File> New notebook ، يمكنك بسهولة إنشاء دفتر ملاحظات جديد من Colab في هذه الصفحة..**



قيود تعلم الآلة

أحدث التعلم الآلي تحولًا في العالم كما نعرفه على مدار العقد الماضي. أدى انفجار البيانات إلى جمع كميات هائلة من البيانات، خاصة من قبل الشركات الكبيرة مثل Facebook و Google. هذا الحجم من البيانات، جنبًا إلى جنب مع التطور السريع لقوة المعالج والتوازي مع أجهزة الكمبيوتر، جعل من الممكن الحصول على كميات كبيرة من البيانات ودراستها بسهولة نسبية. من السهل أن نفهم لماذا كان للتعلم الآلي مثل هذا التأثير العميق على العالم، فما هو أقل وضوحًا هو ما هي قيود التعلم الآلي؟في استمرار لهذا القسم، سوف نوضح هذه القيود.

الاخلاق

ذكر ديفيد بروكس لأول مرة مصطلحًا جديدًا يسمى "**اعتقاد البيانات**[1]" أو اتجاه البياناتفي مقالته عام 2013في صحيفة نيويورك تايمز بعنوان "فلسفة البيانات". اعتقاد البيانات هو نظام أخلاقي تمت دراسته على نطاق واسع ونشره من قبل المؤرخ الشهير يوفال نوح هراري. وفقًا لهراري، يشير اعتقاد البيانات إلى مرحلة جديدة من الحضارة ندخلها، حيث نعتمد على الخوارزميات والبيانات أكثر من اعتمادنا على حكمنا ومنطقنا. من منظور الإيمان بالبيانات، يبدو أننا نواجه صعوبةفي إدارة البيانات التي ينتجها العالم وفهم عقولنا وأجسادنا. بدلاً من ذلك، يجب أن نترك قراراتنا للخوارزميات التي تعرفنا بشكل أفضل. إنهم يقررون من سيواعدون، وماذا يأكلون، وأين يذهبون. فكرة الوثوق بالبيانات والخوارزميات لها مزايا وعيوب أكثر من حكمنا. من الواضح أننا نستفيد من هذه الخوارزميات، وإلا فإننا لا نستخدمهافي المقام الأول. تسمح لنا هذه الخوارزميات بأتمتة العمليات من خلال إصدار أحكام مستنيرة باستخدام البيانات الموجودة.

[1] dataism

ومع ذلك، يعني هذافي بعض الأحيان استبدال وظيفة شخصية بخوارزمية لها عواقب أخلاقية. علاوة على ذلك، إذا حدث خطأ ما، فمن نلوم؟ أكثر مواضيع المناقشة شيوعًافي الوقت الحالي هي السيارات ذاتية القيادة، كيف يجب أن تتفاعل السيارةفي حالة حدوث تصادم مميت؟ هل يجب أن نختار اتباع إطار أخلاقي معين عند شراء سيارةفي المستقبل؟ على من يقع اللوم إذا قتلت سيارتي شخصًا على الطريق؟ من الواضح، مع ذلك، أن التعلم الآلي لا يمكنه إخبارنا بأي شيء عن القيم المعيارية التي يجب أن نقبلها؛ بمعنى، كيف يجب أن نتصرففي العالمفي موقف معين؟

البيانات

البيانات، هذا هو القيد الأكثر وضوحا. إذا قمت بإطعام نموذج بشكل سيئ، فسوف يعطيك نتائج سيئة. يمكن أن يتجلى ذلك بطريقتين: نقص البيانات ونقص البيانات الجيدة.

نقص البيانات

تتطلب العديد من خوارزميات التعلم الآلي كميات كبيرة من البيانات قبل أن تبدأفي تقديم نتائج مفيدة. وخير مثال على ذلك هو الشبكة العصبية. الشبكات العصبية هي آلات تستهلك البيانات[1] وتتطلب كميات كبيرة من بيانات التدريب. كلما كانت البنية أكبر، زادت الحاجة إلى مزيد من البيانات لتحقيق نتائج مقبولة. غالبًا ما تكون زيادة البيانات هي الحل المفضل.

نقص البيانات الجيدة

لنفترض أنك تعتقد أنه يمكنك الغش من خلال إنشاء عشرات الآلاف من نقاط البيانات المزيفة لوضعها على شبكة عصبية. ماذا يحدث عندما تطعمه للشبكة؟ يتم تدريبها على هذه البيانات ولن تعمل بشكل جيد عند اختبارها على مجموعة بيانات غير مرئية (مجموعة تجريبية). كانت لديك البيانات ولكن جودة البيانات لم تكن عالية. مثلما يمكن أن يؤدي عدم وجود ميزات جيدة إلى ضعف أداء الخوارزمية، فإن عدم وجود بيانات جيدة يمكن أن يحد من إمكانات النموذج الخاص بك. لن تقوم أي شركة بتنفيذ نموذج التعلم الآلي الذي يؤدي أداءً أسوأ من الخطأ على المستوى البشري. وبالمثل، فإن استخدام نموذج تم تدريبه على مجموعة من البياناتفي موقف معين قد لا ينطبق بالضرورة على الموقف الثاني. وأفضل مثال يمكن إعطاؤهفي هذا الصدد هو التنبؤ بسرطان الثدي. تحتوي قواعد بيانات التصوير الشعاعي للثدي على العديد من الصور، لكنها تعاني من مشكلة تسببتفي مشاكل كبيرةفي السنوات الأخيرة: جميع صور الأشعة السينية تقريبًا لنساء بيض. قد لا يبدو هذا مشكلة كبيرة، ولكنفي الواقع، فإن النساء السوداوات أكثر عرضة بنسبة 42في المائة للوفاة من سرطان الثدي بسبب مجموعة واسعة من العوامل التي قد

---

[1] data-eating

تشمل الاختلافات في التشخيص والحصول على الرعاية الصحية*. لذلك، فإن تدريس خوارزمية بشكل أساسي على النساء البيض له تأثير سلبي على النساء السود في هذا الصدد. ما نحتاجه في هذه الحالة بالذات هو المزيد من صور الأشعة السينية للمرضى السود في قاعدة البيانات التدريبية، والمزيد من الميزات المتعلقة بسبب هذه الزيادة بنسبة 42 ٪ في احتمالية أداء خوارزمية التصنيف بشكل أفضل.

قابلية التفسير

قابلية التفسير[1] هي إحدى المشاكل الرئيسية للتعلم الآلي. شركة استشارات ذكاء اصطناعي تحاول تقديم اقتراحات لشركة تستخدم الأساليب الإحصائية التقليدية فقط، إذا كان النموذج غير قابل للتفسير ولا يمكنك إقناع عميلك كيف أدركت أن هذه الخوارزمية قد وصلت إلى هذا القرار، ما مدى احتمالية ثقتك بك وخبرتك؟ يمكن أن تبدو هذه النماذج بعد ذلك عاجزة للغاية، ما لم يكن بالإمكان تفسيرها. لهذا السبب، تعتبر قابلية التفسير ميزة مهمة للغاية يجب أن تتبعها أساليب التعلم الآلي من أجل تطبيقها عمليًا (تم إجراء الكثير من الأبحاث لمقاربة قابلية التفسير)

الشرح مقابل التفسير

من التشخيصات الطبية إلى سيناريوهات الأعمال، تُستخدم نماذج التعلم الآلي لاتخاذ قرارات مهمة. للثقة في الأنظمة التي تعمل مع هذه النماذج، نحتاج إلى معرفة كيف تتنبأ هذه النماذج. هذا هو سبب أهمية الفرق بين النموذج القابل للتفسير والنموذج التوضيحي. تعتمد الطريقة التي ندرك بها النماذج والدرجة التي يمكننا من خلالها فهمها حقًا على ما إذا كانت قابلية الشرح أو قابلة التفسير[2]. في مجال التعلم الآلي والذكاء الاصطناعي، غالبًا ما يتم استخدام الشرح والتفسير بالتبادل. في حين أنهم مرتبطون ارتباطًا وثيقًا، فمن الأفضل فهم اختلافاتهم. لأن، دعنا نرى مدى تعقيدها عندما نبدأ في التعمق أكثر في أنظمة التعلم الآلي.

تعني قابلية الشرح في التعلم الآلي أنه يمكنك شرح ما يحدث في النموذج من الإدخال إلى الإخراج، مما يجعل النموذج شفافًا. بعبارة أخرى، فإن فهم الميزات التي تلعب دورًا في التنبؤ بالنموذج ولماذا يفعلون ذلك هو مفهوم قابل للتفسير.

في المقابل، يتم تعريف قابلية التفسير على أنها الدرجة التي يمكن للإنسان أن يفهم بها سبب القرار أو الدرجة التي يمكن للإنسان أن يتنبأ بها باستمرار بنتيجة نموذج التعلم الآلي. بمعنى آخر، إنها الدرجة التي يمكنك من خلالها توقع ما سيحدث بناءً على التغييرات في المدخلات أو

---

* إذا كنت تريد معرفة المزيد عن هذا ، أوصي بقراءة هذا المقال:
https://news.mit.edu/2019/using-ai-predict-breast-cancer-and-personalize-care-0507

[1] Explanability

[2] interpretability

المعاملات الحسابية. على سبيل المثال، تحتاج السيارة إلى وقود لتتحرك، أي الوقود الذي يدفع المحركات: قابل للتفسير. فهم كيف ولماذا يستهلك المحرك ويستخدم الوقود: يمكن شرحه. باختصار، القابلية للتفسير تعني القدرة على التعرف على آلية دون معرفة السبب بالضرورة. التفسير هو القدرة على شرح ما يحدث بشكل كامل.

باختصار، يمكن للبشر فهم النموذج القابل للتفسير دون أي مساعدة أو تقنية أخرى. بمعنى آخر، لا يمكننا فهم كيفية توقع هذه النماذج إلا من خلال النظرفي معاملات النموذج. **يمكن القول إن النموذج القابل للتفسير يقدم تفسيره.في المقابل، لا يقدم النموذج القابل للشرح تفسيرًا وهو أكثر تعقيداً.** كلما كان النموذج أكثر تعقيداً، قل توضيحيًا وبالتالي أقل قابلية للفهم من قبل البشر، وكلما زادت التقنيات التي يحتاجها لفهم كيفية عمل التنبؤات..

وتجدر الإشارة إلى أن القابلية للتفسير والشرح يُرى أحيانًا جنبًا إلى جنب وينقسمان إلى فئتين من الصندوق الأسود ونماذج الصندوق الشفاف (الصندوق الزجاجي أو الصندوق الأبيض) بناءً على درجة التعقيدفي شرح النماذج. نموذج مع القدرة على شرح نموذج الصندوق الشفاف وفي النقطة المعاكسة هو نموذج الصندوق الأسود. بمعنى آخر، يقف نموذج الصندوق الأسودفي تناقض صارخ مع نموذج الصندوق الشفاف. من هذا المنظور، غالبًا ما تكون نماذج الصندوق الأسود معقدة ويصعب تفسيرها. من ناحية أخرى، فإن نماذج الصناديق الشفافة بسيطة للغاية بحيث يمكن شرح تشغيلها مباشرة.

التعلم الآلي القابل للتفسير

نقول إنه يمكن تفسير ما يمكن فهمه. مع وضع هذافي الاعتبار، نقول إنه يمكن تفسير النموذج إذا كان يمكن للإنسان وحده فهمه دون مساعدة من أسلوب آخر. يمكننا أن ننظر إلى معاملات النموذج أو ملخص النموذج ونفهم بالضبط سبب اتخاذ النموذج للتنبؤ أو بعبارة أخرى قرارًا معينًا. تتضمن أمثلة النماذج القابلة للتفسير أشجار القرار والانحدار الخطي. لأنه يمكننا فحص معاملات النموذج مباشرة واستنتاج كيفية تحويل هذه النماذج لمدخلاتها إلى مخرجات. لذلك، فإن هذه النماذج لا تحتاج إلى شرح ولا تحتاج إلى مزيد من الشرح. باختصار، القابلية للتفسير لا تحتاج إلى شرح.

التعلم الآلي القابل للشرح

يمكن اعتبار نموذج التعلم الآلي على أنه دالة تكون خصائص نموذجها مدخلات والتنبؤات هي مخرجات. الدلة التي يصعب شرحها للبشر تسمى الصندوق الأسود. بمعنى آخر، نحتاج إلى طريقة أو تقنية إضافية حتى نتمكن من النظر إلى الصندوق الأسود وفهم كيفية عمل النموذج. مثال على هذا النموذج هو الغابة العشوائية. ببساطة، تتكون الغابة العشوائية من العديد من أشجار القرار التي يتم فيها أخذ تنبؤات جميع الأشجار الفرديةفي الاعتبار عند إجراء التنبؤ النهائي. لفهم

كيفية عمل الغابة العشوائية، يجب أن نفهم في نفس الوقت كيف تعمل كل الأشجار الفردية. حتى مع وجود عدد قليل من الأشجار، لن يكون هذا ممكنًا.

ترتبط درجة قابلية الشرح ارتباطًا مباشرًا بمدى تعقيدات نموذج الصندوق الأسود. النماذج الأكثر تعقيدًا أقل قابلية للشرح (يصعب شرحها وتتطلب المزيد من العمل لفك تشفيرها). على سبيل المثال، عندما نبدأ في دراسة الخوارزميات مثل الشبكات العصبية العميقة، يصبح الموقف أكثر تعقيدًا. AlexNet ، وهي شبكة عصبية التفافية تستخدم للتعرف على الصور ، لديها 62378344 معامل. بمجرد النظر إلى وزن معاملات هذا النموذج، لا يمكن للبشر فهم كيفية عمله.

لماذا يعتبر الشرح والتفسير مهمين فى التعلم الآلى ؟

نظرًا لأن مجالات مثل الرعاية الصحية تسعى إلى نشر الذكاء الاصطناعي، أو بشكل أكثر دقة، أنظمة التعلم العميق، حيث تكون أسئلة الشفافية ذات أهمية خاصة، إذا لم نتمكن من تحسين القابلية للتفسير بشكل صحيح وتقديم التفسير في نهاية المطاف في خوارزمياتنا، فإن التأثير المحتمل للذكاء الاصطناعي سيكون على محمل الجد تظهر قدرتك منخفضة الطاقة. ولكن بصرف النظر عن الاعتبارات المهنية التي يجب إجراؤها، هناك أيضًا حجة مفادها أن تحسين القابلية للتفسير والشرح مهم حتى في سيناريوهات الأعمال الأبسط. يمكن أن يساعد فهم كيفية عمل الخوارزمية حقًا في مواءمة أنشطة علماء ومحللي البيانات بشكل أفضل واحتياجاتهم التنظيمية الرئيسية.

يمكن أن تسهل القابلية للشرح فهم الجوانب المختلفة للنموذج وتؤدي إلى رؤى يمكن استخدامها من قبل مختلف أصحاب المصلحة للمساعدة في معالجة الاهتمامات الرئيسية التالية عند نشر منتج أو اتخاذ قرارات بناءً على التوقعات الآلية:

- **الحقيقة:** هل نحن على يقين من أن جميع متغيرات الاهتمام فقط لعبت دورًا في قرارنا؟ هل نحن على يقين من إزالة الأنماط والارتباطات غير الصحيحة من نتائجنا؟
- **القوة:** في حالة وجود بيانات مفقودة أو مشوشة ، هل نحن على يقين من أن هذا النموذج لا يعمل بشكل صحيح؟
- **التحيز:** هل نحن على دراية بأي تحيزات محددة تعاقب مجموعة من الأشخاص بشكل غير عادل ، وإذا كان الأمر كذلك ، فهل يمكننا تحديدها وتصحيحها؟
- **التحسين:** كيف يمكن تحسين النموذج التنبئي؟ كيف ستؤثر بيانات التدريب الإضافية أو مساحة الميزات المتقدمة على النموذج؟ بمعنى آخر ، إذا كنت تعرف سبب وكيفية عمل نموذجك ، فأنت تعرف بالضبط ما تحتاج إلى تعديله وتحسينه.

- **قابلية النقل:** كيف يمكن تطبيق النموذج التنبئي لمجال على آخر؟ ما هي ميزات البيانات والنماذج التي ينبغي تكييفها لهذه القابلية للنقل؟
- **الثقة:** الثقة مهمة جداً في المجالات عالية الخطورة مثل الرعاية الصحية أو المالية. قبل أن يتم استخدام حلول التعلم الآلي والوثوق بها ، يحتاج جميع أصحاب المصلحة إلى فهم كامل لما يفعله النموذج. إذا ادعت أن نموذجك يتخذ قرارات أفضل ويلاحظ أنماطًا لا يراها البشر ، فيجب أن تكون قادرًا على دعمها بالأدلة. سيكون الخبراء في هذا المجال متشائمين بطبيعة الحال بشأن أي تقنية تدعي أنها تفهم المشكلة بشكل أفضل مما يفعلون.
- **المطابقة:** القدرة على شرح النموذج للعلماء وصناع القرار أمر بالغ الأهمية لضمان الامتثال لسياسات الشركة ومعايير الصناعة واللوائح الحكومية. وفقًا للمادة 14 من قانون حماية البيانات الأوروبي (GDPR) ، عندما تستخدم شركة أدوات آلية لصنع القرار ، يجب أن توفر معلومات مفيدة حول المنطق ذي الصلة بالإضافة إلى الأهمية والعواقب المتوقعة لهذه المعالجة. تم سن لوائح مماثلة في جميع أنحاء العالم.

أضافة التعقيد للتعامل مع التعقيد

على الرغم من أننا كبشر نتفوق في العديد من المهام المعرفية ، بما في ذلك التفكير النقدي والإبداع والتعاطف والعقلية ، إلا أننا لسنا ممتازين في إدارة التعقيدات. وجد علماء النفس أن البشر يمكنهم فقط تتبع حوالي 2±7 شيء في ذاكرتهم العاملة. لكن الأجهزة (مثل الكمبيوتر) يمكنها تتبع ملايين ومليارات العناصر (مقيدة فقط بحجم RAM). نظرًا لأن مشكلة الصندوق الأسود هي مجرد مشكلة معقدة ، يمكننا استخدام تحليل الآلة أو خوارزميات التعلم الآلي الأخرى لشرح الصندوق الأسود.

ومع ذلك ، فأنت تضيف خطوة إضافية إلى عملية التطوير. في الواقع ، ربما تضيف عدة خطوات. من وجهة النظر هذه ، يبدو أنك تحاول التعامل مع التعقيد بمزيد من التعقيد ، وهذا صحيح إلى حد ما. ما يعنيه هذا عمليًا هو أننا إذا أردنا أن نكون جادين حقًا بشأن القابلية للتفسير والشرح ، فيجب أن يكون هناك تغيير أوسع في الطريقة التي يتم بها العلم وكيف يعتقد الناس أنه ينبغي القيام به.

أخيرًا ، بينما يصعب على الدماغ البشري تفسير نماذج الصندوق الأسود ، يمكن تفسيرها جميعًا بمساعدة التحليل والخوارزميات. يسمح لنا العدد المتزايد لأساليب التعلم الآلي وأطر العمل التي تم تطويرها في هذا المجال بالنظر داخل الصناديق السوداء وتحويلها إلى صناديق زجاجية. لذلك ، يمكن القول أن "مشكلة الصندوق الأسود" ليست في الحقيقة مشكلة لا يمكن حلها ، وبالتالي التشكيك في قوة هذه النماذج. إن قادة الأعمال الذين يضحون باستخدام التعلم

الآلي بسبب طبيعة الصندوق الأسود للنماذج يتجاهلون بشكل أساسي طريقة فعالة وموثوقة لتحسين قرارات أعمالهم لشيء يمكن حله فقط.

التقنيات المستخدمة لفهم النماذج القابلة للشرح

يمكن التعامل مع القابلية للتفسير بطريقتين:

1. عامة. هذا شرح عام لسلوك النموذج ويعطينا صورة كبيرة للنموذج وكيف تؤثر خصائص البيانات بشكل جماعي على النتيجة.
2. محلي. تخبرنا هذه الطريقة كيف تؤثر الميزات بشكل فردي على النتيجة.

يجب توخي مستوى من الحذر عند استخلاص النتائج باستخدام هذه الأساليب. هذا لأن هذه التقنيات يمكن أن توفر فقط تقريبًا لكيفية توقع النموذج بالفعل. لتأكيد كل استنتاج، يمكن استخدام العديد من التقنيات معًا أو يمكن تأكيدها باستخدام تصور البيانات. يمكن أن تكون معرفة المجال أيضًا أداة مهمة، ويجب تحليل أي نتيجة تتعارض مع الخبرة أو المعرفة السابقة بمزيد من التفصيل.

ما هو نهج التعلم الآلى الذى يجب أن نستخدمه؟

على الرغم من مناهج التعلم المختلفة (التعلم الخاضع للإشراف ، والتعلم غير الخاضع للإشراف، والتعلم المعزز) والخوارزميات المختلفة ، فإن السؤال هو كيفية تحديد النهج الذي يجب استخدامه لحل مشكلة معينة؟

تتمثل إحدى الإستراتيجيات في تجربة جميع مناهج التعلم الآلي الممكنة ثم التحقق من النهج الأفضل. تكمن مشكلة هذه الطريقة في أنها قد تستغرق وقتًا طويلاً. هناك العشرات من خوارزميات التعلم الآلي ، ولكل منها وقت تشغيل مختلف. اعتمادًا على مجموعة البيانات ، قد تستغرق بعض الخوارزميات ساعات أو حتى أيام حتى تكتمل. من المخاطر الأخرى لتنفيذ استراتيجية "جرب جميع المناهج" أنه قد ينتهي بك الأمر باستخدام خوارزمية التعلم الآلي لنوع من المشكلات غير مناسب حقًا لتلك الخوارزمية المعينة. إنه مثل استخدام مطرقة لتشديد المسمار. المطرقة هي بالتأكيد أداة مفيدة ، ولكن فقط عند استخدامها للغرض المقصود منها. إذا كنت تريد إحكام ربط البرغي ، فاستخدم مفكًا وليس مطرقة.

عند تحديد نوع خوارزمية التعلم الآلي المراد استخدامها ، يجب أولاً فهم المشكلة تمامًا ثم تحديد ما تريد تحقيقه. فيما يلي إطار عمل مفيد يمكن استخدامه لتحديد الخوارزمية المناسبة:

- **هل ترغب في تقسيم مجموعة بيانات غير مسماة إلى مجموعات بحيث تتمتع كل مجموعة بخصائص متشابهة (على سبيل المثال ، تقسيم العملاء)؟** إذا كانت الإجابة

بنعم ، فاستخدم خوارزمية التجميع (التعلم غير الخاضع للإشراف) مثل k-mean أو التجميع الهرمي أو النماذج المختلطة غاوسيان.

- **هل تريد توقع قيمة مستمرة بناءً على مجموعة من الميزات (مثل توقعات سعر السكن)؟** إذا كانت الإجابة بنعم ، استخدم خوارزمية الانحدار (التعلم الخاضع للإشراف) مثل الانحدار الخطي.
- **هل تريد توقع فئات منفصلة؟ هل لدينا مجموعات بيانات تم تمييزها بالفعل بالفئات؟** إذا كانت الإجابة بنعم على كلا السؤالين ، فاستخدم خوارزمية تصنيف (التعلم الخاضع للإشراف) مثل مصنف Naive Bayes أو KNN أو الشبكات العصبية أو آلات المتجهات الداعمة.
- **هل تحاول تقليل عدد الميزات إلى عدد أقل من الميزات؟** استخدم خوارزميات تقليل الأبعاد ، مثل تحليل المكونات الرئيسية PCA.
- **هل تحتاج إلى خوارزمية تستجيب لبيئتها وتتعلم باستمرار من التجربة ، كما يفعل البشر؟** إذا كانت الإجابة بنعم ، استخدم نهج التعلم المعزز.

لكل من الأسئلة أعلاه، يمكنك طرح الأسئلة التالية للتحقق من الخوارزمية المناسبة للاستخدام. علي سبيل المثال:

- **هل نحتاج إلى خوارزمية يمكن بناؤها وتدريبها واختبارها بسرعة؟**
- **هل نحتاج إلى نموذج يمكنه التنبؤ بسرعة؟**
- **ما مدى دقة النموذج؟**
- **هل عدد الميزات أكبر من عدد العينات؟**
- **هل نحتاج إلى نموذج يسهل تفسيره؟**
- **ما هي معايير التقييم المهمة لتلبية احتياجات العمل؟**
- **ما مقدار معالجة البيانات التي نريد القيام بها؟**

خلاصة الفصل الرابع

- التعلم الآلي هو طريقة تعمل على تحسين أداء النظام من خلال التعلم التجريبي من خلال الأساليب الحسابية.
- في أنظمة الكمبيوتر ، توجد الخبرة في شكل بيانات ، وتتمثل المهمة الرئيسية للتعلم الآلي في تطوير خوارزميات التعلم التي تقوم بنمذجة البيانات.
- في النهج التعلم الخاضع للاشراف ، يتم تغذية مجموعة من عينات التدريب بالإجابات الصحيحة للخوارزمية وتحاول الخوارزمية تعلم دالة بناءً على هذه البيانات والإجابات الصحيحة حتى تتمكن من التنبؤ بشكل صحيح بالقيم المستهدفة للعينات الجديدة.

- في التعلم الخاضع للاشراف، إذا كانت بيانات مشكلة التعلم متقطعة ، فهي مشكلة التصنيف ، وإذا كانت قيم البيانات مستمرة ، فإنها تسمى الانحدار.
- في النهج غير الخاضع للاشراف، لا يتم توفير الإجابات الصحيحة للخوارزمية ، ولكن بدلاً من ذلك تحاول الخوارزمية تحديد أوجه التشابه بين المدخلات بحيث يتم تجميع المدخلات التي لها ميزة مشتركة معًا.
- في التعلم المعزز ، يحاول الوكيل حل مشكلة عن طريق التجربة والخطأ من خلال التفاعل مع بيئة طبيعتها غير معروفة للوكيل.
- يركز التعلم الانتقالي على استخراج البيانات من مجال مشابه لزيادة القدرة على التعلم أو تقليل عدد العينات المصنفة المطلوبة في المجال المستهدف.
- يعد التعلم المتعدد المهام نموذجًا تعليميًا يتم من خلاله تعلم العديد من المهام ذات الصلة في وقت واحد من أجل زيادة كفاءة النموذج إلى الحد الأقصى ، كما يتم تحسين العديد من دوال الخطأ في وقت واحد.
- التعلم بدون عينات هو طريقة تعلم خاضعة للاشراف ولكن بدون بيانات تدريبية من هذه الفئة.
- التعلم بدون عينات قادر على حل مشكلة ما دون تلقي أي تعليمات إرشادية من تلك المشكلة.
- التعلم النشط هو فرع من فروع التعلم الآلي حيث يمكن لخوارزمية التعلم التواصل مع المستخدم لتمييز البيانات بالمخرجات المرغوبة.
- الغرض من التعلم النشط هو زيادة أداء خوارزمية التعلم الآلي مع الحفاظ على عدد العينات التعليمية ثابتًا.
- الاستراتيجيتان الأساسيتان في التعلم النشط هما أخذ عينات عدم اليقين وأخذ عينات من مساحة.
- في خوارزميات التعلم الجماعي ، تكون البيانات الإرشادية متاحة بالكامل من البداية إلى وكيل التعلم.
- في خوارزميات التعلم المتزايد ، قد لا تكون بيانات التدريب معروفة أو مكتملة من البداية أو يمكن إضافتها بمرور الوقت.
- تعمل خوارزميات التعلم الآلي على أتمتة عملية تعلم النموذج ، مما يوضح العلاقة بين الخصائص الوصفية وخاصية الهدف في مجموعة البيانات.
- الهدف من التعلم الآلي هو العثور على النموذج التنبئي الأكثر عمومية.
- يستخدم التعلم الآلي والتقيب في البيانات نفس التقنيات وتتداخل فيما بينها.

مصادر إضافية لمزيد من القراءة

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. T. (2012). Learning from data (Vol. 4, p. 4). New York, NY, USA:: AMLBook.

Alpaydin, E. (2020). Introduction to machine learning. MIT press.

Brink, H., Richards, J., & Fetherolf, M. (2016). Real-world machine learning. Simon and Schuster.

Chen, L. P. (2019). Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar: Foundations of machine learning.

Flach, P. (2012). Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press.

Marsland, S. (2011). Machine learning: an algorithmic perspective. Chapman and Hall/CRC.

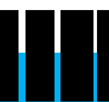
Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

Theodoridis, S. (2015). Machine Learning: A Bayesian and Optimization Perspective. .NET Developers Series.

Watt, J., Borhani, R., & Katsaggelos, A. K. (2020). Machine learning refined: Foundations, algorithms, and applications. Cambridge University Press.

5

اختيار النموذج وتقييمه



الأهداف:

- الفرق بين النموذج والخوارزمية.
- التعرف على مفاهيم التحيز والتباين.
- طرق التقييم.
- طرق ضبط المعاملات الفائقة.
- تقييم الأداء.

النموذج والخوارزمية

هناك ارتباك شائع للمبتدئين في التعلم الآلي وهو الفرق بين "خوارزمية التعلم الآلي" و "نموذج التعلم الآلي". غالبًا ما يتم استخدام المصطلحين بالتبادل ، مما قد يكون مربكًا. باختصار ، إنهما ليسا متماثلين ، فإن خوارزمية التعلم الآلي تشبه الإجراء الذي يتم تشغيله على البيانات للعثور على الأنماط والقواعد المخزنة والمستخدمة لإنشاء نموذج للتعلم الآلي. في هذا القسم، سنخبرك ما هو الفرق بين الخوارزميات والنماذج في التعلم الآلي؟

الخوارزمية

خوارزمية التعلم الآلي هي إجراء يتم تنفيذه على البيانات لإنشاء "نموذج" للتعلم الآلي. بمعنى آخر، تؤدي خوارزمية التعلم الآلي "التعرف على الأنماط" و "التعلم" من البيانات. لتبسيط الموضوع ، يمكن إظهار العلاقة بينهما على النحو التالي:

نموذج التعلم الآلي ← خوارزمية التعلم الآلي

هناك أنواع مختلفة من الخوارزميات بوظائف وأغراض مختلفة. الوظائف الثلاثة الرئيسية هي:

- **الانحدار:** للتنبؤات حيث يكون الناتج قيمة مستمرة.
- **التصنيف:** يستخدم للتنبؤات التي يكون ناتجها قيمة مصنفة.
- **التجميع:** لتجميع الأشياء المتشابهة أو نقاط البيانات في مجموعات.

عندما تقوم بتدريب "خوارزمية" بالبيانات، فإنها تصبح "نموذجًا".

تشبه خوارزميات التعلم الآلي أي خوارزمية أخرى في علوم الكمبيوتر. وبالتالي ، فإن خوارزميات التعلم الآلي لها الخصائص التالية:

- يمكن وصف خوارزميات التعلم الآلي باستخدام الرياضيات وشبيه الكود (سودوكود).
- يمكن تحليل أداء خوارزميات التعلم الآلي ووصفها.
- يمكن تنفيذ خوارزميات التعلم الآلي مع أي من لغات البرمجة الحديثة.

على سبيل المثال، قد ترى خوارزميات التعلم الآلي ذات الشبيه كود أو الرياضيات في المقالات والكتب المدرسية. قد ترى الأداء الحسابي لخوارزمية تعلم آلة معينة مقارنة بخوارزمية أخرى معينة. يمكن للباحثين تطوير خوارزميات جديدة تمامًا للتعلم الآلي ، ويمكن لمتخصصي التعلم الآلي استخدام خوارزميات التعلم الآلي القياسية في مشاريعهم. هذا تمامًا مثل أي مجال آخر في علوم الكمبيوتر حيث يمكن للباحثين اختراع خوارزميات فرز جديدة تمامًا ، على سبيل المثال ، ويمكن للمبرمجين استخدام خوارزميات الفرز القياسية في تطبيقاتهم.

النموذج

"النموذج" في التعلم الآلي هو ناتج خوارزمية التعلم الآلي التي تعمل على البيانات وتمثل ما تم تعلمه بواسطة خوارزمية التعلم الآلي. لإظهار العلاقة بينهما ، يمكننا استخدام العلاقة التالية:

نموذج التعلم الآلي= بيانات النموذج + خوارزمية التنبؤ

أخيرًا ، النموذج هو "شيء" يتم تخزينه على بيانات التدريب بعد تنفيذ خوارزمية التعلم الآلي ويعرض القواعد والأرقام وهياكل البيانات المحددة الأخرى للخوارزمية اللازمة للتنبؤ.

"نموذج" التعلم الآلى هو المكان الذى يتم فيه تخزين ناتج "الخوارزمية". يمكن حفظ النموذج لوقت لاحق ويعمل كبرنامج، باستخدام الوظيفة المحفوظة مسبقًا للخوارزمية للتنبؤات الجديدة. إذا تم تدريب النموذج بكفاءة وبشكل كافٍ، فيمكن استخدامه لعمل تنبؤات إضافية حول بيانات مماثلة بمستوى معين من الدقة والموثوقية.

الاختلافات بين النماذج والخوارزميات فى التعلم الآلى

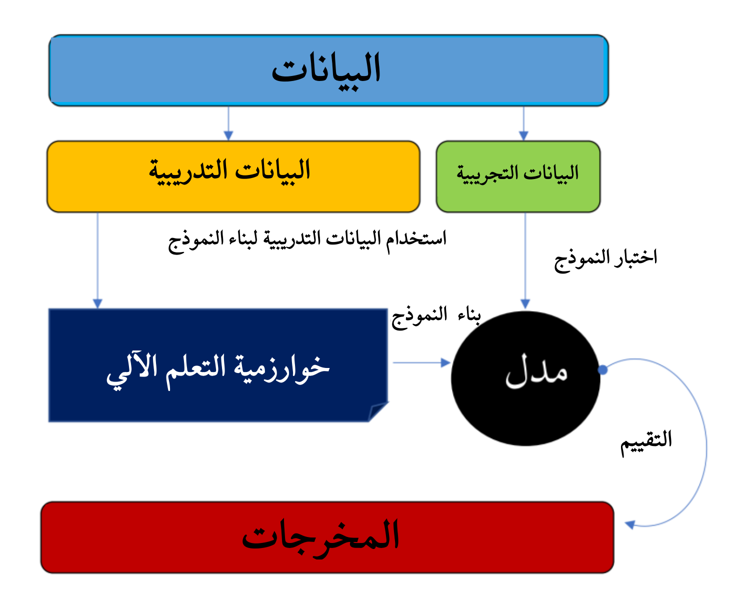
الآن بعد أن عرفنا ما هي الخوارزمية والنموذج ، أصبح من السهل معرفة كيفية ارتباطهما. كما ذكرنا سابقًا ، يتم تشغيل خوارزمية على البيانات لإنشاء نموذج. يتكون هذا النموذج من بيانات وطريقة لاستخدام البيانات للتنبؤ ببيانات جديدة. هذه الطريقة تشبه خوارزمية التنبؤ تقريبًا. ومع ذلك ، لا تخزن جميع النماذج خوارزمية تنبؤ. البعض ، مثل KNN ، يخزن مجموعة البيانات بأكملها ، والتي تعمل كخوارزمية تنبؤ. نحن أساسًا نريد "نموذجًا" للتعلم الآلي ولا نهتم بالخوارزمية التي تقف وراءه. بمعنى آخر ، الخوارزمية هي مجرد المسار الذي نتبعه للحصول على النموذج. ومع ذلك ، من المهم معرفة الخوارزمية التي يجب تطبيقها على نموذجك للحصول على أفضل النتائج. بمجرد أن تعرف هذا ، لا يوجد سوى بضعة أسطر من التعليمات البرمجية وعدد قليل من مستويات التفاعل حتى تتمكن من الحصول على نموذج فعال بالكامل. باختصار ، يمكن تحديد الفرق بين النموذج والخوارزمية في التعلم الآلي على النحو التالي:

- خوارزميات التعلم الآلي هي إجراءات يتم إجراؤها على البيانات للعثور على الأنماط والتعلم.
- نماذج التعلم الآلي هي نتاج الخوارزميات وتتكون من بيانات وخوارزمية تنبؤ.
- توفر خوارزميات التعلم الآلي نوعًا من البرمجة الآلية حيث تمثل نماذج التعلم الآلي البرنامج نفسه.

نموذج التعلم الآلى هو برنامج تتم كتابته أو إنشاؤه أو تعلمه تلقائيًا بواسطة خوارزمية التعلم الآلى لحل مشكلتنا. بصفتنا مطورين، فإننا أقل اهتمامًا بـ "التعلم" الذى تقوم به خوارزميات التعلم الآلى بمعنى الذكاء الاصطناعى. نحن لا نهتم بمحاكاة عمليات التعلم. قد يجد بعض الناس هذا مثيرًا للاهتمام، لكن هذا ليس سبب استخدامنا لخوارزميات التعلم الآلى. بدلاً من ذلك، نحن مهتمون أكثر بقدرات البرمجة الذاتية التى توفرها خوارزميات التعلم الآلى. نريد إنشاء نموذج فعال حتى نتمكن من دمجه فى مشروعنا البرمجى. تقوم خوارزميات التعلم الآلى بالبرمجة الذاتية، ونماذج التعلم الآلى هى برامج تم إنشاؤها لنا.

الإطار العام لخوارزميات التعلم الآلى

يتمثل الإطار العام لخوارزميات التعلم الآلي في بناء نموذج لتوليد مخرجات لمدخلات جديدة لم تتم ملاحظتها بعد، باستخدام مجموعة من البيانات. يمكن أن يشتمل إطار التعلم، الموضح في الشكل 5ــ1، على: البيانات (التدريب والتجارب)، والخوارزمية المستخدمة لبناء النموذج، وأخيرًا تقييم النموذج لإنتاج المخرجات، وسيتم شرح كل منها في الأقسام التالية.



الشكل 5ــ1 الإطار العام لخوارزميات التعلم الآلي

تطوير نموذج

خوارزمية التعلم ودالة الخطأ

خوارزمية التعلم A هي تعيين مجموعة بيانات محدودة S إلى الدالة المرشحة $\hat{f}$ حيث تكون $\hat{f}$ قابلة للقياس. نفترض أن بياناتنا x_i, y_i موزعة بشكل مستقل وموحد من مساحة الاحتمال $X \times Y$ بالحجم ρ. نحدد معنى "**الأداء الجيد**"في مهمة ما عن طريق إدخال دالة الخطأ، دالة قابلة للقياس $L: X \times Y \times F \to [0, \infty)$. تكون هذه الخطأ دائمًا تقريبًافي شكل $L(x, y, f) = \acute{L}(y, f(x))$ لبعض دوال $\acute{L}$. ومن ثم، سوف نستخدم طريقة الكتابة هذه فيما يلي. بطبيعة الحال، يجب أن نعتبر $L(y, \hat{f}(x))$ مقياسًا لتكلفة التنبؤ لـ $\hat{f}(x)$، حيث y هي التسمية الحقيقية لـ x. إذا توقعنا أن $f(x) = y$ ، فإننا نقول إن تنبؤنا عند x ممتاز (لا تشوبه شائبة) ونتوقع ألا نعاني من أي خطأ عند x (على سبيل المثال $L\left(y, \hat{f}(x)\right) = 0$). يعد اختيار دالة الخطأ الصحيحة جزءًا مهمًا من استخدام التعلم الآليفي الممارسة. فيما يلي، نقدم عينات على المهام ذات مساحات البيانات المختلفة X, Y ودوال الخطأ المختلفة L.

مثال 1: تصنيف الصور

تصنيف الصورة يتعلق الأمر بتصنيف الصورة x إلى إحدى الفئات المحتملة C. هنا $x \epsilon R^{H \times W \times 3}$ حيث H و W هما ارتفاع الصورة وعرضها و 3 هي قنوات الألوان (الأحمر والأخضر والأزرق). مساحة التسمية محدودة لمجموعة محدودة $Y = C$. ينتج ناتج نموذج المصنف توزيعًا متقطعاً $f(x_i) = (p_1, \ldots, p_c)$ على الفئات ، حيث تتوافق p_c مع احتمال أن تكون صورة الإدخال x من الفئة c. كدالة خطأ ، نستخدم **خطأ الانتروبيا المتقاطعة**[1] لهذه المشكلة:

$$L\big(y, f(x)\big) = -\frac{1}{N}\sum_{c=1}^{C} 1\{y_i = c\} \log(p_c)\,, \quad p = f(x_i)$$

مثال 2: توقع خصائص الكريستال

من الممارسات الشائعة في علم المواد التنبؤ بخصائص البلورة (على سبيل المثال ، طاقة التكوين) من هيكلها الذري (رسم بياني بلا اتجاه). كمشكلة تعلم ، إنها مشكلة انحدار مع X

---

[1] cross-entropy loss

كمجموعة من الرسوم البيانية بلا اتجاه و $Y = \mathbb{R}$. بالنسبة لدالة الخطأ ، من الشائع استخدام متوسط الخطأ المطلق (MAE) بسبب ثباته بالنسبة إلى النقاط البعيدة:

$$L(y, f(x)) = |y - f(x)|$$

مشكلة التعلم

يشير التعلم الآلي إلى عملية استخدام الأدوات الإحصائية لتعلم البيانات وفهمها. التعلم يدور حول إيجاد الدالة $\hat{f}$ ، والتي تمتد من البيانات المحدودة S إلى المساحة اللانهائية $X \times Y$. يمكن التعبير عن هذه الفكرة على أنها تقليل الخطأ المتوقع، وتسمى أيضًا المخاطر[1]:

$$\mathcal{E}(f) = \mathbb{E}\left[L(y, f(x))\right] = \int_{X \times Y} L(y, f(x)) d_\rho(x, y)$$

هدفنافي التعلم هو تقليل المخاطر:

$$f^* = argmin_{f \in F} \mathbb{E}\left[L(y, f(x))\right]$$

$$= argmin_{f \in F} \int_{X \times Y} L(y, f(x)) d_\rho(x, y)$$

نظرًا لأن لدينا بيانات محدودة، فمن المستحيل أيضًا حساب المخاطر. بدلاً من ذلك، نقوم بتقريبها باستخدام بياناتنا وننشئ **مخاطر تجريبية**[2]:

$$\hat{\mathcal{E}}(f) = \frac{1}{n} \sum_{i=1}^{N} L(y_i, f(x_i)) \approx \int_{X \times Y} L(y, f(x)) d_\rho(x, y) \qquad 1\text{–}2$$

هذا المفهوم، المسمى **بتقليل المخاطر التجريبية**، هو أساس التعلم الآلي الحديث.

يمكن أن نأمل أنه من خلال تقليل المخاطر التجريبية على جميع الدوال القابلة للقياس، يمكننا تقريب التعبير الموجود على اليمين 2–1 والحصول على الدالة $\hat{f} = argmin_{f \in F} \hat{\mathcal{E}}(f)$ مشابهة لـدالة الحالة f^* . ومع ذلك، فإن هذا غير ممكن بدون افتراضات أو شروط مسبقة إضافية.في هذه الإعدادات غير المقيدة، لا يمكن لأي نموذج تحقيق خطأ منخفض في جميع توزيعات البيانات، وهي نتيجة لا تُعرف باسم **نظرية الغداء ليس بالمجان**[3].

---

[1] Risks

[2] empirical risk

[3] No Free Lunch Theorem

يسمى الفرق بين أداء دالة التعلم التجريبية $\hat{f}$ وأفضل أداء ممكن **بفجوة التعميم أو خطأ التعميم**[1]. هدفنا هو تقليل احتمالية حدوث هذا الخطأ:

$$\mathbb{P}(\widehat{\mathcal{E}}(f) - inf_{f\in F}\varepsilon(f) > \epsilon)$$

إذا تم تقليص هذه الفجوة بشكل لا نهائي إلى الصفر، فالمطلوب هو:

$$\lim_{n\to\infty} \mathbb{P}\left(\widehat{\mathcal{E}}(f) - inf_{f\in F}\varepsilon(f) > \epsilon\right) \quad = 0 \ \forall_{\epsilon} > 0$$

يسمى الخطأ التجريبي أحيانًا خطأ التعميم. هذا لأنه، في الواقع، في معظم الحالات، ليس لدينا وصول إلى مجال X بأكمله للمدخلات، ولكن فقط إلى مجموعة التدريب الفرعية S، ونريد التعميم بناءً على S، والذي يسمى أيضًا التعلم الاستقرائي[2].

الغداء ليس بالمجان!

نظرية الغداء المجاني (NFL)في التعلم الآلي الخاضع للإشراف هي نظرية تعني أساسًا أنه لا توجد خوارزمية للتعلم الآلي هي أفضل خوارزمية بشكل عام لكل شيء. قد يبدو غريباً بعض الشيء، لكن الفكرة التي ربما ألهمت نظرية NFL اقترحها فيلسوف لأول مرةفي القرن الثامن عشر. نعم، لقد قرأت ذلك بشكل صحيح! ليس عالم رياضيات أو إحصائيًا، بل فيلسوف.في منتصف القرن الثامن عشر الميلادي، اقترح الفيلسوف الأسكتلندي ديفيد هيوم ما أسماه **مشكلة الاستقراء**[3]. هذا سؤال فلسفي يسأل عما إذا كان الاستدلال الاستقرائي يقودنا حقًا إلى المعرفة الحقيقية. الاستدلال الاستقرائي هو شكل من أشكال التفكير الذي نستخلص فيه الاستنتاجات بناءً على الملاحظات السابقة للكون. والمثير للدهشة أن هذا هو بالضبط ما تفعله خوارزميات التعلم الآلي. إذا شاهدت الشبكة العصبية 100 صورة لبجعة بيضاء، فمن المحتمل أن تستنتج أن جميع البجع أبيض. لكن ماذا يحدث إذا رأت الشبكة العصبية بجعة سوداء؟ يتم الآن رفض النمط الذي تعلمته الخوارزمية بشكل مفاجئ من خلال مثال متبادل واحد. غالبًا ما يشار إلى هذه الفكرة باسم **مفارقة البجعة السوداء**[4].

استخدم هيوم هذا المنطق لتسليط الضوء على قيود الاستدلال الاستقرائي الذي لا يمكننا تطبيق الاستنتاجات حول مجموعة من الملاحظات على مجموعة أكثر عمومية من الملاحظات.

---

[1] generalization error

[2] inductive learning

[3] the problem of induction

[4] the black swan paradox

يقول ديفيد هيومفي رسالة عن الطبيعة البشرية: "*لا يوجد دليل يثبت ذلك، فالأشياء التي لم نختبرها مشابهة لما اختبرناه*". بعد أكثر من 200 عام، ألهمت نظرية NFL التعلم الآلي.

في مقالته[1] عام 1996، قدم ويلبيرت مسألة الغداء المجاني للتعلم الآلي، مستخدمًا الاقتباس من ديفيد هيومفي بداية مقالته. تنص هذه النظرية على أنه بالنسبة لمجموعة البيانات الخالية من الضوضاء، لكل من خوارزميات التعلم الآلي A وB، سيكون متوسط أداء A وB هو نفسهفي جميع عينات المشكلات المحتملة المأخوذة من توزيع الاحتمال الموحد.

لماذا هذا صحيح؟ يعود هذا إلى مفهوم التفكير الاستقرائي. تضع كل خوارزمية للتعلم الآلي افتراضات سابقة حول العلاقة بين السمات والمتغيرات المستهدفة لمشكلة تعلم الآلة. غالبًا ما تسمى هذه الافتراضات الافتراضات السابقة[2]. يعتمد أداء خوارزمية التعلم الآليفي أي مشكلة على مدى توافق افتراضات الخوارزمية مع حقيقة المشكلة. قد تعمل الخوارزمية جيداً لحل مشكلة ما، ولكن لا يوجد سبب للاعتقاد بأنها تعمل جيداً على مشكلة مختلفة حيث قد لا تعمل الفرضيات. هذا المفهوم هوفي الأساس مفارقة البجعة السوداءفي التعلم الآلي.

الافتراضات المقيدة التي تقوم بها عند اختيار أي خوارزمية هي نفس السعر الذي تدفعه مقابل الغداء. هذه الافتراضات تجعل الخوارزمية الخاصة بك أفضل بشكل طبيعيفي بعض القضايا بينمافي نفس الوقت بشكل طبيعي أسوأفي البعض الآخر.

كل هذه النظريات رائعة، ولكن ماذا يعني NFL بالنسبة لك كعالم بيانات أو مهندس تعلم آلي أو مجرد مبتدئ؟ وهذا يعني أن جميع الخوارزميات متساوية؟ بالطبع لا!!في الممارسة العملية، لا يتم إنشاء كل الخوارزميات بنفس الطريقة. هذا لأن المجموعة الكاملة من مشاكل التعلم الآلي هي مفهوم نظريفي حالة NFL وهي أكبر بكثير من مجموعة مشاكل التعلم الآلي العملية التي سنحاول حلها بالفعل. قد تؤدي بعض الخوارزميات عادةً أداءً أفضلفي أنواع معينة من المشكلات أكثر من غيرها، ولكن لكل خوارزمية مزايا وعيوب بسبب الافتراضات السابقة التي قدمتها تلك الخوارزمية. ربما تكون خوارزمية مثل XGBoost قد فازت بالمئات من سباقات Kaggle، ولكن بسبب الافتراضات المحدودةفي النماذج القائمة على الأشجار، فقد فشلت فشلاً ذريعًافي التنبؤ بالمهام. قد تؤدي الشبكات العصبية أداءً جيداًفي المهام المعقدة مثل تصنيف الصور والتعرف على الكلام، ولكن إذا لم يتم تدريبها بشكل صحيح، فقد تعاني من الضبط الزائد[3] بسبب تعقيدها.

من الناحية العملية ، هذا ما يعنيه NFL:

---

[1] The Lack of A Priori Distinctions Between Learning Algorithms

[2] priori assumptions

[3] Overfitting

- لا توجد خوارزمية واحدة تحل جميع مشكلات التعلم الآلي أفضل من أي خوارزمية أخرى.
- قبل اختيار خوارزمية لاستخدامها ، تأكد من فهمك الكامل لمشكلة التعلم الآلي وبياناتها.
- تتميز النماذج الأبسط ، مثل الانحدار اللوجستي ، بانحياز أعلى وتميل إلى الضبط الناقص ، بينما تتمتع النماذج الأكثر تعقيداً ، مثل الشبكات العصبية ، بمزيد من التباين وتميل إلى الضبط الزائد.
- أفضل النماذج لمشكلة معينة تقع في مكان ما بين حافتي التباين_المتحيز.
- للعثور على نموذج جيد لمشكلة ما ، قد تضطر إلى تجربة نماذج مختلفة ومقارنتها باستخدام استراتيجية تحقق قوية.

التنبؤ مقابل الاستدلال

بشكل عام، الهدف من أي عملية تعلم خاضعة للإشراف هو توقع متغير كمي أو نوعي y بناءً على مجموعة من p المتنبئين $x_1, x_2 \dots, x_p$. بالإضافة إلى ذلك، نفترض أن هناك علاقة بين y و x_p.في أبسط الحالات، يمكن التعبير عن هذه العلاقة على النحو التالي:

$$y = f\left(x_p\right) + \epsilon$$

حيث f دالة غير معروفة ولكن مع الثوابت $x_1, x_2 \dots, x_p$ و ϵ هو تعبير خطأ يمثل جميع المتغيرات التي تتعلق بـ y ولكنها غير مضمنةفي النموذج.في الإحصاء، نفترض أن تعبير الخطأ مستقل عن x_p وأن متوسطه هو صفر. بمعنى آخر ، f هي دالة (غير معروفة) ترسم العلاقة بين متغير الاستجابة والمتنبئ. نظرًا لأن الدالة غير معروفة، يتعين علينا تقدير هذه الدالة بناءً على نقاط البيانات المرصودة.في التحليل الإحصائي، هناك مجالان رئيسيان للاهتمام لتقدير دالة f، وهما **الاستدلال والتنبؤ**.

يهتم التعلم الآلي بشكل أساسي بالتنبؤ. على وجه الخصوص، فهي مهتمة بتحديد مجموعة المتنبئين التي توفر أكثر التنبؤات دقة لمخرجات y وهي أقل اهتمامًا بطبيعة العلاقة. بمعنى آخر، طالما أن قوة التوقع عالية ومتسقة، فليس من الضروري وجود علاقة سببية بين متنبئ x_p ومتغير الاستجابة y. نتيجة لذلك، لا نحتاج إلى وضع افتراضات حول البيانات والشكل الدقيق لـ f.
نظرًا لأن بيان الخطأ هو صفرفي المتوسط ، يمكننا التنبؤ بـ y بناءً على هذه المجموعة من المتنبئين.

$$\hat{y} = \hat{f}(x)$$

حيث $\hat{f}$ هي دالة تقدير f و $\hat{y}$ هي القيم المتوقعة لـ y. في التعلم الآلي، يكون الهدف عادةً هو تقدير الدالة f ، مما يقلل خطأ التنبؤ.

تعريف التنبؤ

استخدم النموذج للتنبؤ بنقاط البيانات الجديدة

يشير التنبؤ ، عند استخدامه كأسم ، إلى بيانات قاطعة حول حدث أو حادث فى المستقبل. قد يستند تقييمها أو لا يستند إلى البيانات والحقائق والأدلة. ليس من الواضح دائمًا ما إذا كان التنبؤ سيكون صحيحًا أم لا. هذا لأن التنبؤ بالمستقبل غير معروف. عند استخدامه كفعل ، فإن المصطلح يعنى "التنبؤ". ومن الأمثلة على ذلك أن عالم الأرصاد الجوية " تنبؤ" ما إذا كانت ستمطر أم لا. أو "تنبأ" العراف ببيع المنزل قريباً. بعض الأمثلة على مصطلح "التنبؤ" فى جملة واحدة هى: " تنبؤه" عن المستقبل كان خاطئًا. أو "تنبؤ" على أن يفوز الفريق الأحمر بالبطولة فى ذلك اليوم.

تعريف الاستنتاج

استخدام النموذج للتعرف على عملية توليد البيانات

يشير الاستنتاج ، عند استخدامه كأسم ، إلى فعل الوصول إلى نتيجة تم تقييمها على أساس البيانات والحقائق والأدلة المتاحة. يتضمن ذلك بناء نموذج يصف العلاقة بين المتغيرات ونتائج حدث أو حدث باستخدام البيانات الإحصائية. نظرًا لأن التقييم حقيقى ، فهناك الكثير من اليقين. بالإضافة إلى ذلك ، قد لا يدور الاستنتاج بالضرورة حول مستقبل غير معروف. يستخدم مصطلح " الاستنتاج" عادة عندما يتعلق الاستنتاج بالحاضر. عند استخدامه كفعل ، يُعرف المصطلح باسم "استنتاج". هذا يعنى التوصل إلى نتيجة. على سبيل المثال ، إذا كان مظهر الأطفال سيئًا عندما يأكلون طعامًا ، فإن أمهاتهم "يستنتجن" أنهم لا يحبون الطعام. أو إذا أعطى الناس رأيًا سلبيًا حول مطعم ما ، "يُستنتج" أن طعامهم سيء. بعض الأمثلة على مصطلح "الاستنتاج" فى جملة واحدة هى: "استنتج" على ما كان تحت الطاولة ، أو طلب المحقق من مساعده "الاستنتاج" بناءً على القرائن المتاحة.

فى التنبؤ، نحن مهتمون بتقدير f بأكبر قدر ممكن من الدقة حتى نتمكن من توقع المتغير المستهدف Y بناءً على المتغيرات المستقلة X. فى الاستنتاج، من ناحية أخرى، ما زلنا مهتمين بتقدير f ، لكن هذه المرة ليس لعمل تنبؤات ، ولكن لفهم العلاقة بين X و Y.

التفسير هو ضرورة للاستنتاج

بشكل أساسي، يتم تقليل الاختلاف بين النماذج المناسبة للاستنتاج والنماذج غير المناسبة إلى قابلية تفسير النموذج. ما المقصود بقابلية تفسير النموذج؟ يمكن تفسيره على أنه نموذج يمكن للإنسان من خلاله إعادة النظرفي كيفية إنشاء تقديراته. ضعفي اعتبارك الطرق التالية للتنبؤ:

- **قابل للتفسير:** النماذج الخطية المعممة (مثل الانحدار الخطي والانحدار اللوجستي) وتحليل الفصل الخطي[1] وآلات متجه الدعم الخطي وأشجار القرار.
- **غير قابل للتفسير (أقل قابلية للتفسير):** الشبكات العصبية وآلات الدعم غير الخطية والغابات العشوائية.

فقط مجموعة فرعية من الطرق القابلة للتفسير مفيدةفي الاستنتاج. على سبيل المثال، يمكن تفسير آلات متجه الدعم الخطي لأنها توفر معاملًا لكل خاصية بحيث يمكن تفسير تأثير الخصائص الفردية على التنبؤ. ومع ذلك، لا تسمح آلات المتجهات الداعمة بتقدير حالات عدم اليقين المتعلقة بمعاملات النموذج (مثل التباين) ولا يمكن الحصول على مقياس ضمني لموثوقية النموذج. لاحظ أن آلات المتجه الداعم قادرة على توليد احتمالات الإخراج، ولكن هذه الاحتمالات ليست سوى تحويل لقيم القرار ولا تستند إلى موثوقية تقدير المعاملات. هذا هو السببفي أن حتى الطرق القابلة للتفسير مثل آلات متجهات الدعم الخطي وأشجار القرار غير كافية للاستنتاج.في المقابل، ضعفي اعتبارك الانحدار الخطي، والذي يفترض أن البيانات تتبع توزيع غاوسي. تحدد هذه النماذج الخطأ المعياري لتقدير معاملات وفترات موثوقية المخرجات. نظرًا لأن الانحدار الخطي يسمح لنا بفهم الطبيعة المحتملة لعملية إنتاج البيانات، فهي طريقة جيدة للاستنتاج. تحظى الطرق البايزية للاستدلال بشعبية كبيرة لأنه يمكن تكييف هذه النماذج لتجمع بين افتراضات مختلفة حول عملية إنتاج البيانات.

فقط لأن لديك نموذجًا مفيدًا للاستدلال لا يعني أنك تفعل ذلك حقًا. ما يهم هو كيفية استخدام النموذج. على سبيل المثال، على الرغم من أن النماذج الخطية المعممة مناسبة للاستدلال، إلا أنه لا يمكن استخدامها إلا للأغراض التنبؤية. ضعفي اعتبارك الأمثلة التالية التي تجعل التمييز بين التنبؤ والاستدلال أكثر وضوحًا:

- **التنبؤ:** تريد التنبؤ بمستويات الأوزون المستقبلية باستخدام البيانات السابقة. نظرًا لأنك تعتقد أن هناك علاقة خطية بين مستوى الأوزون وقياس درجة الحرارة والإشعاع الشمسي والرياح، فإنك تضع عدة نماذج خطية على بيانات التدريب وتختار النموذج الذي يقلل الخطأفي المجموعة التجريبية. أخيرًا، يمكنك استخدام النموذج المحدد للتنبؤ بمستويات الأوزون.

[1] Linear Discriminant Analysis (LDA)

لاحظ أنه طالما أن النموذج يقلل من خطأ الاختبار، فأنت لا تهتم على الإطلاق بافتراض نموذج كاوسيان أو المعلومات الإضافية الواردةفي تقدير النموذج.

- **الاستدلال:** تريد أن تفهم كيف يتأثر سطح الأوزون بدرجة الحرارة والإشعاع الشمسي والرياح. يمكنك استخدام نموذج الانحدار الخطي لأنك تفترض أن البيانات يتم توزيعها بشكل طبيعي. يتم استخدام مجموعة البيانات الكاملة للنمذجة من أجل الحصول على المعلومات وبما أن دقة التنبؤ ليست مهمة بالنسبة لك. بناءً على النموذج المناسب، يمكنك تفسير دور الخصائص على مستوى الأوزون المُقاس. على سبيل المثال، النظرفي تقديرات نطاقات الثقة (فترات الثقة).

تجزئة البيانات

عند تنفيذ نماذج التعلم الآلي للأغراض التنبؤية، من المهم تقسيم البيانات بشكل صحيح لتقييم أداء النماذج بشكل موضوعي ولتجنب أوجه القصورفي البيانات.في الحالات التي توجد فيها بيانات كافية، يمكن تقسيم البيانات بشكل عشوائي إلى ثلاث مجموعات: مجموعة التدريب ومجموعة التحقق ومجموعة الاختبار. تستخدم مجموعة التدريب، كما يوحي اسمها، لتدريب النماذج وتناسبها. تُستخدم مجموعة التحقق من الصحة للحصول على قيم المعاملات الفائقة المثلى (تحسين المعاملات الفائقة) وللمساعدةفي تحديد النموذج، ويتم استخدام مجموعة الاختبار لتقييم أداء النموذج النهائيفي العينات التي تظهرفي عملية التعلم. لا توجد قواعد واضحة حول حجم مجموعات مختلفة من مجموعات البيانات، لأن هذا يعتمد إلى حد كبير على توافر البيانات. ومع ذلك، عادةً ما تكون مجموعة التدريب هي الجزء الأكبر من البيانات، حيث يجب تدريب نماذج التعلم الآلي على كميات كبيرة من البيانات لتكون فعالة.

اختيار النموذج وتقييمه

يدور التعلم الآلي حول مفهوم الخوارزميات أو النماذج التي تقوم بالفعل بتقديرات إحصائية. ومع ذلك ، فإن كل نموذج له العديد من القيود اعتمادًا على توزيع البيانات. لا يمكن أن يكون أي منها دقيقًا تمامًا ، لأنها مجرد تقديرات. تُعرف هذه القيود عمومًا باسم التحيز والتباين. يتم تبسيط نموذج التحيز العالي من خلال عدم الانتباه إلى نقاط التدريب (على سبيل المثال ، في الانحدار الخطي ، بغض النظر عن توزيع البيانات ، يعتبر النموذج دائمًا علاقة خطية). أيضًا ، يحد نموذج التباين العالي نفسه بالبيانات التدريبية من خلال عدم تعميم نقاط الاختبار التي لم يرها من قبل.

المتعلمون الجيدون الذين نبحث عنهم هم الذين يؤدون أداءً جيدًا في الأمثلة الجديدة. لذلك، يجب أن يتعلم المتعلمون الجيدون القواعد العامة من الأمثلة التدريبية بحيث تنطبق القواعد المستفادة على جميع الحالات المحتملة (غير المرئية). ومع ذلك ، عندما يتعلم المتعلم الأمثلة

التدريبية "جيدًا جدًا" ، فمن المحتمل أن يتم اعتبار بعض ميزات العينة التدريبية كميزات عامة ستتمتع بها جميع الحالات المحتملة ، مما يؤدي إلى انخفاض الأداء العام. في التعلم الآلي ، تُعرف هذه الظاهرة بالضبط الزائد وعكسه الضبط الناقص ، أي يفشل المتعلم في تعلم السمات العامة للأمثلة التدريبية.

تعريف الضبط الزائد Overfitting

يشير الضبط الزائد إلى الموقف الذى يتتبع فيه النموذج الإحصائى بدقة عالية الضوضاء أو الأخطاء بدلاً من اتباع إشارة فى البيانات.

إذا كان نموذج التعلم الآلى يعمل بشكل جيد للغاية فى بيانات التدريب (خطأ تدريب منخفض) ولكنه يعمل بشكل ضعيف عند اختبار بيانات جديدة (خطأ اختبار مرتفع) ، فهذا عادة ما يكون علامة على أن النموذج Overfitting. هذا وضع سيء للغاية ، لأنه يعنى أن النموذج متسق للغاية مع بيانات التدريب وغير قادر على تعميم علاقات السمات على البيانات الجديدة. هذا لأن النموذج يتذكر البيانات التى شاهدها ولا يمكنه تعميمها على العينات غير المرئية.

فى التعلم الخاضع للإشراف ، يحدث الضبط الزائد عندما يلتقط نموذجنا الضوضاء جنبًا إلى جنب مع النمط الأساسى فى البيانات. على عكس الضبط الزائد ، فى التعلم الخاضع للإشراف ، يحدث الضبط الناقص عندما يفشل النموذج فى الحصول على النمط الأساسى للبيانات.

من بين الأسباب العديدة المحتملة ، تعد القدرة على التعلم بقوة مفرطة سببًا شائعًا للضبط الزائد ، حيث يمكن لمثل هؤلاء المتعلمين تعلم السمات غير العامة للأمثلة التدريبية. في المقابل، عادة ما يكون الضبط الناقص بسبب ضعف القدرة على التعلم. في الممارسة العملية ، من السهل نسبيًا التغلب على الضبط الزائد. على سبيل المثال ، يمكننا القيام بالمزيد من التفرع في تعلم أشجار القرار أو إضافة المزيد من الدورات التدريبية في تعلم الشبكات العصبية. ومع ذلك ، كما سنرى لاحقًا ، يمثل الضبط الزائد مشكلة كبيرة في التعلم الآلي ، وبالتالي ، تم تنفيذ طرق مختلفة لتقليلها. ومع ذلك ، يجب أن نعلم أن الضبط الزائد أمر لا مفر منه وأن كل ما يمكننا فعله هو تقليله وليس القضاء عليه تمامًا. يمكن تبرير هذه الحجة بإيجاز على النحو التالي:

غالبًا ما تكون مشكلات التعلم الآلي NP-hard أو أكثر صعوبة ، ولكن يجب أن تكمل خوارزميات التعلم العملية التعلم في وقت متعدد الحدود. لذلك ، إذا كان من الممكن الضبط الزائد ، فإن تقليل الخطأ التجريبي يؤدي إلى الحل الأمثل ، وبالتالي لدينا دليل بناء على $P=NP$. بعبارة أخرى ، طالما أننا نؤمن بـ $P \neq NP$ ، فإن الضبط الزائد أمر لا مفر منه.

من الناحية العملية ، غالبًا ما يكون هناك العديد من خوارزميات التعلم المرشحة ، وحتى خوارزمية تعلم واحدة قد تنتج نماذج مختلفة وفقًا لمعايير مختلفة. إذن ما هي خوارزمية التعلم التي يجب أن نختارها وما هي إعدادات المعاملات التي يجب استخدامها؟ هذا يسمى اختيار النموذج. الحل المثالي هو تقييم جميع النماذج المرشحة واختيار النموذج الأقل خطأ في التعميم. ومع ذلك ، لا يمكننا الحصول على خطأ التعميم مباشرة ، بينما يعاني الخطأ التجريبي من الضبط الزائد. إذن ، كيف يمكننا تقييم واختيار النماذج عمليًا؟ في ما يلي ، بعد مراجعة موجزة لتوازن التحيز والتباين ، سنصف طرق ومعايير اختيار النموذج.

توازن التحيز والتباين[1]

الغرض الرئيسي من نموذج التعلم الآلي هو التعلم من البيانات التي تم تغذيتها وعمل تنبؤات بناءً على النمط الذي لوحظ أثناء عملية التعلم. ومع ذلك ، هذه ليست نهاية مهمتنا. يتعين علينا إجراء تحسينات باستمرار على النماذج ، بناءً على نوع النتائج التي تنتجها. نحدد أداء النموذج باستخدام معايير مثل الدقة ومتوسط الخطأ التربيعي ودرجة F1 وما إلى ذلك ، ونحاول تحسين هذه المعايير. غالبًا ما يصبح هذا مشكلة عندما نحتاج إلى الحفاظ على مرونة النموذج. لأن أداء نموذج التعلم الآلي يتم تقييمه بناءً على درجة دقة توقعه ودرجة تعميمه على مجموعة بيانات مستقلة أخرى لم يراها في عملية التعلم.

يعتزم نموذج التعلم الآلي الخاضع للإشراف تدريب نفسه على متغيرات الإدخال X بحيث تكون القيم المتوقعة Y أقرب ما يمكن إلى القيم الفعلية. هذا هو الفرق بين القيم الفعلية وقيم الخطأ المتوقعة ويستخدم لتقييم النموذج. بشكل عام ، يمكن تقسيم خطأ أي خوارزمية للتعلم الآلي إلى فئتين:

1. تقليل الأخطاء.
2. أخطاء لا رجوع فيها.

الأخطاء التي لا رجعة فيها هي أخطاء لا يمكن تقليلها حتى مع أي نموذج آخر للتعلم الآلي. على سبيل المثال، الضجيج خطأ لا رجوع فيه ولا يمكننا القضاء عليه. من ناحية أخرى، يعد التحيز والتباين أخطاء قابلة للقياس يمكننا محاولة تقليلها قدر الإمكان. بسبب هذا التباين–التحيز، يؤدي نموذج التعلم الآلي بالبيانات المقدمة إلى الضبط الزائد والضبط الناقص. يتطلب الحد من الأخطاء اختيار النماذج التي تتمتع بالتعقيد والمرونة المناسبين. يحتاج علماء البيانات إلى فهم الفرق بين التحيز والتباين بشكل كامل لتقليل الأخطاء وبناء نماذج دقيقة.

[1] Bias-Variance Trade-Off

التحيز هو عدم قدرة نموذج التعلم الآلي على الحصول على علاقة حقيقية بين متغيرات البيانات. هذا بسبب الافتراضات الخاطئة داخل خوارزمية التعلم. على سبيل المثال،في الانحدار الخطي، يُفترض أن تكون العلاقة بين X والمتغير Y خطية، بينمافي الواقع قد لا تكون هذه العلاقة خطية تمامًا.في المقابل، على عكس التحيز، يكون التباين عندما يأخذ النموذجفي الاعتبار تقلبات البيانات، بمعنى آخر، الضوضاء. لذلك، عندما يحتوي نموذجنا على تباين كبير، أي أن النموذج يتعلم الكثير من بيانات التدريب (التباين ليس سوى مفهوم ضبط زائد للنموذجفي مجموعة بيانات معينة)، بحيثفي مواجهة البيانات الجديدة (التجارب))، فهو غير قادر على التنبؤ بدقة.

تعريف التحيز

التحيزهو الفرق بين متوسط توقع نموذجنا والقيمةَ الصحيحةَ التى نحاول توقعها.

لا يولى النموذج ذو التحيز العالى اهتماماً كبيرًا ببيانات التدريب ويجعل النموذج بسيطًا للغاية ويؤدى دائمًا إلى خطأ كبير فى بيانات التدريب والاختبار.

تعريف التباين

التباين هو تغيرفى توقع النموذج لنقطةَ بيانات معينةَ أو قيمةَ توضح لنا تشتت البيانات.

يولى نموذج التباين العالى الكثير من الاهتمام لبيانات التدريب ولا يعمم على البيانات التى لم يراها من قبل. نتيجة لذلك ، تعمل هذه النماذج جيدًا على بيانات التدريب ، ولكن لديها معدل خطأ مرتفع فى البيانات التجريبية.

يوضح التباين مدى تغير تقدير الدالة الموضوعية إذا تم استخدام بيانات تدريب مختلفة. بمعنى آخر ، يعبر التباين عن مدى اختلاف متغير عشوائى عن قيمته المتوقعة. يمكن أن يؤدى التباين إلى الضبط الزائد، حيث تزداد التقلبات الصغيرة فى مجموعة التدريب. قد يعكس النموذج ذو التباين العالى ضوضاء عشوائية فى مجموعة بيانات التدريب بدلاً من الدالة الموضوعية. يؤدى النموذج ذو التباين العالى إلى تغييرات كبيرة فى تنبؤات الدالة الهدف.

عند بناء خوارزمية تعلم آلي خاضعة للإشراف ، فإن الهدف هو تحقيق انحياز وتباين منخفضين للحصول على أكثر التنبؤات دقة. قد يمثل نموذج التباين العالي مجموعة البيانات بدقة ، ولكن يمكن أن يؤدي إلى الضبط الزائد. في المقابل ، قد لا يكون نموذج التحيز العالي مناسبًا لبيانات التدريب. يعتمد تحدي التوازن على نوع النموذج المعني. تُظهر خوارزمية تعلم الآلة الخطية انحيازًا مرتفعًا ولكن تباينًا منخفضًا. من ناحية أخرى ، ستظهر الخوارزمية غير الخطية انحيازًا منخفضًا ولكن تباينًا كبيرًا. يؤدي استخدام نموذج خطي مع مجموعة بيانات غير خطية إلى إدخال تحيز في النموذج. هذا النموذج غير مناسب لدوال الهدف مقارنة بمجموعة

بيانات التدريب. العكس صحيح أيضًا ، إذاكنت تستخدم نموذجًا غير خطي على مجموعة بيانات خطية ، فسيكون النموذج غير الخطي متسقًا جدًا مع الدالة الهدف.

لمواجهة تحديات التوازن هذه ، يجب على عالم البيانات تطوير خوارزمية تعلم مرنة بما يكفي للتكيف بشكل صحيح مع البيانات. ومع ذلك ، هناك دائمًا توتر بين التحيز والتباين. في الواقع، من الصعب إنشاء نموذج يحتوي على كل من التحيز والتباين المنخفض (لا مفر من العلاقة بين التحيز والتباين في التعلم الآلي):

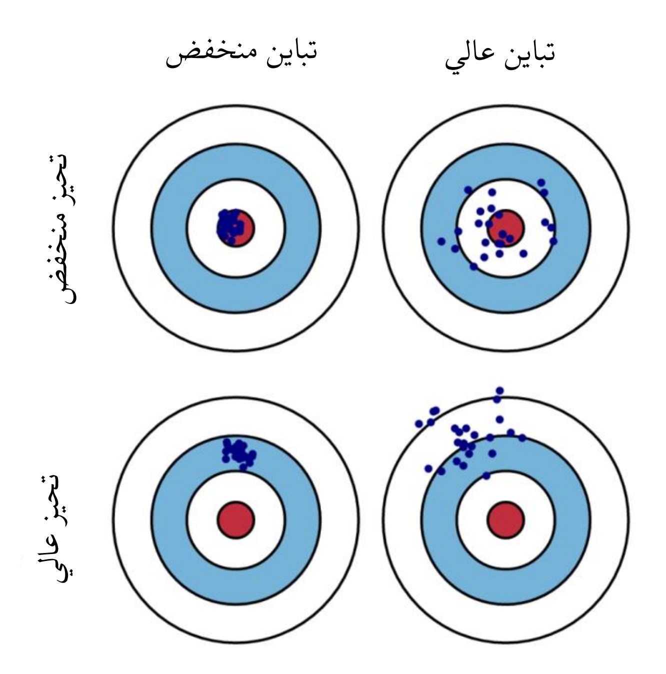
زيادة التحيز يقلل التباين.

زيادة التباين يقلل التحيز.

يعد توازن التحيز ــ التباين مشكلة خطيرة في التعلم الآلي. هذا هو الموقف الذي لا يمكنك فيه الحصول على كل من التحيز المنخفض والتباين المنخفض. في الواقع ، لا يمكننا حساب التحيز والتباين الفعليين لأننا لا نعرف دالة الهدف الرئيسية الفعلية. ومع ذلك ،كإطار عمل ، يوفر التحيز والتباين أدوات لفهم سلوك خوارزميات التعلم الآلي في تتبع الأداء التنبئي ، ويجب تعليمك نموذجًا دقيقًا بدرجة كافية وقابل للتعميم على الأنظمة الحالية في مجموعة مختلفة من البيانات. يحصل على نقاط من مصدر واحد ، ولديه توازن مع التحيز والتباين الأمثل.

التحيز والتباين خطأان فى الخطأ الكلى فى خوارزمية التعلم ، إذا حاولت تقليل خطأ واحد، فقد يزيد خطأ آخر.

توضح الصورة أدناه تباين ــ التحيز بشكل أفضل. مركز النتيجة هو النموذج الذي نريد تحقيقه والذي يتنبأ بجميع القيم بشكل صحيح. عندما نبتعد عن المركز ، يبدأ نموذجنا في عمل تنبؤات خاطئة أكثر فأكثر. يتنبأ النموذج ذو التحيز المنخفض والتباين العالي بالنقاط الموجودة بشكل عام حول المركز ، ولكنها متباعدة جدًا. النموذج ذو التحيز العالي والتباين المنخفض بعيد عن المركز ، ولكن نظرًا لأن التباين منخفض ، فإن النقاط المتوقعة تكون أقرب إلى بعضها البعض. بمعنى آخر ، إذا زاد التباين ، تكون البيانات أكثر انتشارًا ، مما يؤدي إلى دقة أقل (كما هو موضح في الدائرة اليمنى العلوية في الصورة). على العكس من ذلك ، إذا زاد التحيز ، يزداد الخطأ المحسوب (كما هو موضح في الدائرة اليسرى السفلية في الصورة).



عندما تحتوي خوارزمية التعلم على مشكلة تحيز عالية ، فإن العمل على تقليل التحيز يؤدي إلى زيادة التباين ويخلق مشكلة الضبط الزائد، وعندما تعاني خوارزمية التعلم من مشكلة تباين عالية ، فإن العمل على تقليل التباين يؤدي إلى زيادة التحيز وزيادة المشكلة تخلق الضبط الناقص. وهنا يأتي دور مصطلح "التوازن" ، لأن تقليل التحيز وحده لا يؤدي إلى تحسين النموذج ، والعكس صحيح. "النقطة المثلى" هي وضع نقاط البيانات في مكان يوجد فيه تحيز أمثل وتباين أمثل. بشكل أساسي ، ابحث عن نمط من خلال عدم استخدام أي من عناصر الأطراف التي تغير الدقة. غالبًا ما يكون تخطيط واختيار هذه النقاط هو التحدي الأكبر الذي يواجهه علماء البيانات والمحللون. ومع ذلك ، هناك طرق لاختبار ملاءمة النموذج. بعض الحلول المقدمة للتعامل مع هذه الظاهرة هي:

- **بناء نموذج أكثر تعقيداً:** الحل الأول والأسهل لمشكلة الضبط الناقص هو تعليم نموذج أكثر تعقيداً لحل المشكلة ، وإدخال المزيد من البيانات للنموذج الذي يعاني من الضبط الناقص.
- **ضوضاء الانحدار[1]:** تتضمن هذه الطريقة إضافة ضوضاء متدرجة أثناء التدريب ، وهي طريقة ثبت أنها تزيد من دقة النموذج *.

---

[1] Gradient Noise

* انظر هذا المقال:

Adding Gradient Noise Improves Learning for Very Deep Networks: https://arxiv.org/abs/1511.06807

- **التنظيم:** طريقة شائعة للحد من ظاهرة الضبط الزائد. تتضمن هذه التقنية ، التي تُستخدم لحل مشكلة التباين العالي ، الضبط الدقيق للمعاملات والأوزان لتحقيق دقة أعلى لكل من بيانات التدريب وبيانات الاختبار.
- **التسوية:** يحل المشاكل المتعلقة بالإفراط في التركيب والتركيب.

باختصار:

- التحيز هو افتراض مبسط يقوم به النموذج (لأنه يتجاهل البيانات) لتسهيل تقريب دالة الهدف.
- يتسبب النموذج الذي به خطأ تباين كبير في الضبط الزائد ويتعلم منه الكثير.
- إذا كنت تستخدم نموذجًا بسيطًا للتعلم الآلي ، فسيكون للنموذج انحياز كبير وتباين منخفض (الضبط الناقص).
- إذاكنت تستخدم نموذجًا متطورًا للتعلم الآلي ، فسيكون له تباين كبير وانحياز منخفض (الضبط الزائد).
- تحتاج إلى إيجاد توازن جيد بين التحيز والتباين في النموذج الذي تستخدمه. هذا التوازن هو ما يُعرف باسم توازن التحيزـــالتباين.

طرق التقييم

بشكل عام ، يمكننا تقييم خطأ التعميم من خلال التجارب على بيانات الاختبار. للقيام بذلك ، نستخدم مجموعة اختبار لتقدير قدرة المتعلم على تصنيف عينات جديدة ، ونستخدم خطأ الاختبار كتقريب لخطأ التعميم. هنا ، نأخذ في الاعتبار خطأ التعميم فقط ، ولكن في تطبيقات العالم الواقعي ، غالبًا ما نفكر في المزيد من العوامل مثل التكلفة الحسابية وتكلفة الذاكرة وما إلى ذلك. عادة ، نفترض أن عينات الاختبار مستقلة. وتجدر الإشارة إلى أن مجموعة الاختبار ومجموعة التدريب يجب أن تكون محددة بقدر الإمكان ، أي يجب ألا تظهر عينات الاختبار في مجموعة التدريب أو تُستخدم بأي شكل من الأشكال في عملية التدريب.

لماذا لا تظهر عينات الاختبار في البرنامج التدريبي؟ لفهم هذا ، دعنا نفكر في السيناريو التالي. لنفترض أننا نستخدم مجموعة من عشرة أسئلة لكلٍ من الممارسة والامتحان ، فهل يعكس الاختبار نتائج تعلم الطلاب؟ الجواب "لا". لأن بعضَ الطلاب يمكنهم الحصول على درجات جيدة حتى لو تمكنوا من حل هذه الأسئلة العشرة فقط. وبالمثل ، فإن القدرة القابلة للتعميم التي نرغب في تصميمها هي ما نريد من الطلاب دراسته وإتقانه. وعليه تكون عينات التدريب مع التمارين والعينات التجريبية مطابقة للاختبار. ومن ثم ، يمكن أن يكون التقدير متفائلاً للغاية إذا كانت عينات الاختبار قد تم تدريبها بالفعل في عملية التدريب.

ومع ذلك ، بالنظر إلى مجموعة واحدة فقط من البيانات $D = \{(x_1, y_1), (x_2, y_2), \dots (x_m, y_m)\}$من العينات ، كيف يمكننا القيام بكل من التدريب والاختبار؟ الإجابة هي إنشاء مجموعة تدريب S ومجموعة T تجريبية من مجموعة بيانات D. سوف نصف بعض الأساليب الشائعة المستخدمة في هذا الصدد أدناه.

طريقة الصيانة [1]

تقسم طريقة الصيانة مجموعة البيانات D إلى مجموعتين فرعيتين منفصلتين: واحدة كمجموعة تدريب S والأخرى كمجموعة تجريبية T حيث:

$$S \cap T = \emptyset \text{ و } D = S \cup T$$

نقوم بتدريب نموذج على مجموعة التدريب S ثم نحسب خطأ الاختبار على المجموعة التجريبية T كتقدير عام للخطأ. على سبيل المثال ، بالنظر إلى مشكلة التصنيف الثنائي ، افترض أن D هي مجموعة بيانات تحتوي على 1000 عينة ، والتي نقسمها إلى مجموعة تدريبية S بها 700 عينة ومجموعة تجريبية T مع 300 عينة. بعد التدريب على S ، افترض أن النموذج صنف 90 عينة بشكل غير صحيح في T ، لذلك لدينا معدل خطأ 30%=100% × (300/90) وبالتالي فإن الدقة 70%=30%-1.

تجدر الإشارة إلى أن التقسيم يجب أن يحافظ على توزيع البيانات الأصلي لمنع التحيز الإضافي. على سبيل المثال ، عند التفكير في مشكلة التصنيف ، يجب أن نحاول الحفاظ على نسبة الفئة في الفئات الفرعية المختلفة. تسمى طرق أخذ العينات التي تحافظ على نسب الفئة **بأخذ العينات المصنفة**[2]. على سبيل المثال ، لنفترض أن لدينا مجموعة بيانات D تحتوي على 500 عينة موجبة و 500 عينة سلبية ، ونريد تقسيمها إلى مجموعة التدريب S مع 70٪ من العينات والمجموعة التجريبية T مع 30٪ من العينات. ومن ثم ، فإن طريقة أخذ العينات المصنفة تضمن أن يحتوي S على 350 عينة إيجابية و 350 عينة سلبية وأن T يحتوي على 150 عينة إيجابية و 150 عينة سلبية. بدون أخذ العينات الفئوية ، يمكن أن تؤدي نسب الفئات المختلفة في S و T إلى تقديرات خطأ التحيز[3] لأن توزيعات البيانات تتغير. ومع ذلك ، تؤدي طرق التصنيف المختلفة إلى مجموعات تدريبية وتجريبية مختلفة ، وبالتالي ستؤدي إلى نتائج تقييم مختلفة. ومن ثم ، ينتج عن اختبار واحد عادة تقدير خطأ غير موثوق به.

---

[1] Hold-out Method

[2] Stratified sampling

[3] biased error estimation

من الناحية العملية ، غالبًا ما نجري اختبار صيانة عدة مرات ، حيث يتم تقسيم البيانات عشوائيًا في كل اختبار ويستخدم متوسط الخطأ كتقدير نهائي. على سبيل المثال ، يمكننا تقسيم مجموعة البيانات عشوائيًا 100 مرة لإنتاج 100 نتيجة تقييم ثم اعتبار المتوسط بمثابة تقدير لخطأ الصيانة.

تقسم طريقة الصيانة D إلى مجموعة تدريب ومجموعة تجريبية ، لكن النموذج الذي نريد تقييمه هو النموذج الذي يتم تدريبه على D. لذلك نحن أمام معضلة. إذا وضعنا معظم الأمثلة في مجموعة التدريب S ، فإن النموذج المدرب يعد تقديرًا ممتازًا للنموذج المدرب على D. ومع ذلك ، فإن التقييم أقل موثوقية نظرًا لصغر حجم T. من ناحية أخرى ، إذا وضعنا المزيد من العينات في مجموعة اختبار T ، فإن الفرق بين نموذج التدريب الذي يتم تدريبه في S ونموذج التدريب الذي يتم تدريبه في D يصبح كبيرًا ، أي أن دقة التقييم أقل. لا يوجد حل مثالي لهذه المشكلة وعلينا أن نحقق التوازن. من الممارسات الشائعة استخدام $\frac{2}{3}$ الى $\frac{4}{5}$ عينة للتدريب والباقي للاختبار.

طريقة التحقق المتبادل [1]

يقسم التحقق المتبادل مجموعة بيانات D إلى k مجموعة فرعية من نفس الحجم ، أي:

$$D = D_1 \cup D_2 \cup \cdots \cup D_K , D_i \cap D_j = \emptyset \ (i \neq j)$$

عادة ، تحاول كل مجموعة فرعية D_i الحفاظ على توزيع البيانات الأصلي من خلال أخذ العينات المصنفة. هناك عدة طرق لمشاركة البيانات للتحقق المتبادل. في طريقة التحقق من الصحة متعددة الأجزاء، يتم استخدام المجموعة الفرعية K-1 كمجموعة تدريب لتعليم نموذج والمجموعة الفرعية المتبقية كمجموعة تجريبية لتقييم النموذج. كرر هذه العملية K مرة واستخدم كل مجموعة فرعية كمجموعة اختبار مرة واحدة بالضبط. أخيرًا ، للحصول على نتيجة التقييم ، نقوم بمتوسط اختبار K. القيمة الأكثر شيوعًا المستخدمة لـ K هي 10. تتضمن قيم K الشائعة الأخرى 5 و 20.

في تقنية التحقق من الصحة متعددة الأجزاء ، يتم وضع كل جزء من البيانات مرة واحدة بالضبط في مجموعة تجريبية ومرة واحدة في مجموعة التدريب. هذا يقلل بشكل كبير من التحيز والتباين ، لأنه يضمن أن كل مثيل من مجموعة البيانات الأصلية لديه فرصة للظهور في مجموعة التدريب والتجريبية. يعد التحقق متعدد الأجزاء أحد أفضل الطرق لتقييم أداء النموذج إذا كانت لدينا بيانات إدخال محدودة.

[1] Cross-Validation

Bootstrapping

ما نريد تقييمه هو النموذج الذي تم تدريبه باستخدام D. ومع ذلك ، حتى إذا استخدمنا تقنية الصيانة أو التحقق المتبادل ، فستظل مجموعة التدريب أصغر من D. ومن ثم ، فإن تقدير التحيز[1] أمر لا مفر منه بسبب اختلاف الحجم بين مجموعة التدريب و D. bootstraping هو حل يستخدم تقنية أخذ عينات bootstraping. بالنظر إلى مجموعة D تحتوي على عينات m ، يختار التمهيد للمجموعة D'بشكل عشوائي عينة من D وينسخها إلى D' ثم يضعها في D بحيث لا يزال أمامها فرصة للاختيار في المرة القادمة.. يؤدي تكرار هذه العملية مع عدد مرات إلى ظهور مجموعة بيانات أخذ عينات D' Bootstrap مجموعه تحتوي على عينات m. بسبب الاستبدال، قد لا تظهر بعض العينات في D في D'. ، بينما قد تظهر عينات أخرى أكثر من مرة. إذا قمنا بتقدير سريع: فرصة عدم الاختيار في m تساوي $(1-\frac{1}{m})^m$، لذلك لدينا حد:

$$\lim_{m\to\infty}\left(1-\frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368$$

هذا يعني أن ما يقرب من 36.8٪ من العينات الأصلية لا تظهر في مجموعة بيانات D. ومن ثم، يمكننا استخدام D' كمجموعة تدريب و $D\backslash D'$ كمجموعة تجريبية ، بحيث يستخدم كل من النموذج المقيَّم والنموذج الفعلي الذي نريد تقييمه على D عينات التدريب m. بالإضافة إلى ذلك ، لا يزال لدينا مجموعة منفصلة من الاختبارات تحتوي على حوالي ثلث العينات الأصلية التي لم يتم استخدامها للتدريب. تسمى نتيجة التقييم التي تم الحصول عليها من هذه الطريقة **تقدير خارج الحقيبة**[2].

يكون Bootstraping مفيداً عندما تكون مجموعة البيانات صغيرة أو عندما لا توجد طريقة فعالة لتقسيم مجموعات التدريب والمجموعات التجريبية. بالإضافة إلى ذلك ، يمكن لعملية التمهيد إنشاء مجموعات بيانات متعددة يمكن أن تكون مفيدة لطرق مثل التعلم الجماعي. ومع ذلك ، نظراً لتغير توزيع البيانات الأصلية باستخدام Bootstraping ، فإن التقدير متحيز أيضًا. لذلك ، عندما يكون لدينا الكثير من البيانات ، غالبًا ما يتم استخدام تحقق الصيانة والتحقق المتبادل بدلاً من ذلك.

ضبط المعاملات الفائقة والنموذج النهائي

تحتوي معظم خوارزميات التعلم على معاملات فائقة لضبطها ، وغالبًا ما يؤدي تعديل المعاملات الفائقة المختلفة إلى نماذج ذات أداء مختلف. لذلك ، فإن تقييم النموذج واختياره

[1] estimation bias

[2] out-of-bag estimate

لا يتعلق فقط باختيار خوارزميات التعلم ، بل يتعلق أيضًا بتعيين المعاملات الفائقة. تسمى عملية إيجاد المعاملات الفائقة الصحيحة **ضبط المعاملات الفائقة**[1]. تعد المعاملات الفائقة وسيطات نموذجية يتم تعيين قيمها قبل بدء عملية التعلم. مفتاح خوارزميات التعلم الآلي هو تحديد المعاملات. بمعنى آخر ، ضبط المعاملات الفائقة هو عملية تحديد التركيبة الصحيحة من المعاملات الفائقة التي تسمح للنموذج بزيادة أداء النموذج إلى الحد الأقصى. يعد ضبط التركيبة الصحيحة للمعاملات الفائقة الطريقة الوحيدة لاستخراج أقصى أداء من النماذج.

قد يعتقد القراء أنه لا يوجد فرق جوهري بين إعداد معامل فائق واختيار خوارزمية: كل ضبط للمعامل الفائق يؤدي إلى نموذج ، ونختار النموذج الذي ينتج أفضل النتائج كنموذج نهائي. هذه الفكرة صحيحة في الأساس. ومع ذلك ، هناك مشكلة ، نظرًا لأن المعاملات غالبًا ما تكون قيمًا حقيقية[2] ، فمن المستحيل اختبار جميع إعدادات المعاملات الفائقة. وبالتالي ، في الممارسة العملية ، نحدد عادةً نطاقًا وحجم خطوة لكل معامل فائق. على سبيل المثال ، النطاق [0 ، 0.2] وحجم الخطوة هو 0.05 ، مما ينتج عنه خمسة إعدادات للمعاملات الفائق المرشحة فقط. مثل هذا الموازنة[3] بين التكلفة الحسابية والجودة المقدرة يجعل التعلم ممكنًا ، على الرغم من أن إعداد المعامل الفائق المختار لا يكون عادةً هو الأمثل. في الواقع ، حتى بعد إجراء مثل هذا الحساب، لا يزال ضبط معامل فائق يمثل تحديًا كبيرًا. يمكننا عمل تقدير بسيط. افترض أن الخوارزمية تحتوي على ثلاثة معاملات فائقة وكل منها يأخذ في الاعتبار خمس قيم مرشحة فقط، لذلك يتعين علينا تقييم $5^3 = 125$ نموذجًا لكل زوج من مجموعات التدريبية والتجريبية. غالبًا ما تحتوي خوارزميات التعلم القوية على عدد كبير جدًا من المعاملات الفائقة لتعديلها ، مما يؤدي إلى عبء عمل ثقيل في تحديد المعاملات.

غالبًا ما تكون جودة ضبط المعاملات الفائقة أمرًا بالغ الأهمية في تطبيقات العالم الحقيقي. ومع ذلك ، فإن اختيار المجموعة الصحيحة من المعاملات الفائقة ليس بالمهمة السهلة. هناك طريقتان رئيسيتان لتعديلهما:

- **الإعداد اليدوي للمعلمات الفائقة:** في هذه الطريقة ، يتم ضبط واختبار مجموعات مختلفة من المعاملات الفائقة يدويًا. هذه عملية شاقة وقد لا تكون عملية في الحالات التي يوجد فيها عدد كبير جدًا من المعاملات الفائقة للاختبار. بمعنى آخر ، في كل مرة نقوم فيها باختبار معاملات فائقة مختلفة ، يتعين علينا تدريب نموذج على بيانات التدريب ، والتنبؤ ببيانات التحقق من الصحة ، ثم حساب معايير التحقق من الصحة. هذا يجعل العملية اليدوية غير قابلة للحل مع عدد كبير من المعاملات الفائقة في

[1] Hyperparameter Tuning

[2] real-valued

[3] trade off

النماذج المعقدة مثل التعلم الجماعي أو الشبكات العصبية العميقة التي قد تستغرق عدة أيام!

- **الإعداد التلقائي للمعاملات التشعبية:** في هذه الطريقة ، يتم العثور على المعاملات الفائقة المثلى باستخدام خوارزمية تعمل على أتمتة العملية وتحسينها. البحث العشوائي ، والبحث الشبكي ، وتحسين Bayesian هي أمثلة على الضبط التلقائي للمعامل التلقائي.

البحث الشبكى

في طريقة البحث الشبكي، نقوم بإنشاء شبكة من القيم الممكنة للمعاملات الفائقة. يختبر كل تكرار مجموعة من المعاملات الفائقة بترتيب معين. يناسب هذا النموذج مع أي مجموعة من المعاملات الفائقة الممكنة ويسجل أداء النموذج. أخيرًا، يقوم بإرجاع أفضل نموذج بأفضل معاملات فائقة. نظرًا لأن هذه الطريقة تختبر جميع مجموعات المعاملات الفائقة، فإنها تزيد من التعقيد الزمني للحسابات.

البحث العشوائى

هناك طريقة بسيطة لاستبدال البحث الشبكي عن طريق أخذ عينات عشوائية من مساحة المعامل الفائق. بمعنى آخر ، بدلاً من التجارب المنتظمة على مجموعة القيم الكاملة في مساحة المشكلة، من الأفضل تحديد واختبار القيم العشوائية من مساحة العينة بأكملها. في طريقة البحث العشوائي، نقوم بإنشاء شبكة من القيم الممكنة للمعاملات الفائقة. يختبر كل تكرار مجموعة عشوائية من المعاملات الفائقة من هذه الشبكة ، ويسجل الأداء ، ثم يُرجع في النهاية مجموعة المعالمات الفائقة التي توفر أفضل أداء.

تحسين Bayesian

يعد إعداد وإيجاد المعاملات الفائقة الصحيحة للنموذج مهمة تحسين. بمعنى آخر، نريد تقليل دالة الخطأفي نموذجنا عن طريق تغيير معاملات النموذج الفائقة. يساعدنا تحسين Bayesianفي العثور على الحد الأدنى من النقاطفي أقل عدد من الخطوات. يستخدم تحسين Bayesian أيضًا دالة شاملة توجه أخذ العينات إلى المناطق التي يحتمل أن يكون فيها التحسين على أفضل الملاحظات الحالية. بشكل عام، المفهوم الأساسي لتحسين Bayesian هو: "*إذا بحثنا عشوائيًا عن بعض النقاط ووجدنا أن بعض هذه النقاط واعدة أكثر من غيرها، فلماذا لا ننظر إليها؟* ".

يتطلب تحسين Bayesian عادةً تكرارًا أقل لتحقيق المجموعة المرغوبة من قيم المعاملات الفائقة ، لأنه يتجاهل مناطق مساحة المعامل التى يعتقد أنها لا تساعد على الإطلاق.

كما أن طرق البحث الشبكي والعشوائي غير فعالة نسبيًا، لأنهم لا يختارون المعاملات الفائقة التالية للتقييم بناءً على النتائج السابقة. شبكة البحث والبحث العشوائي غير مدركين تمامًا للتقييمات السابقة ، ونتيجة لذلك ، غالبًا ما يقضون وقتًا طويلاً في تقييم المعلمات الفائقة "السيئة"..

تقييم الأداء

في كل مرة تنشئ فيها نموذجًا للتعلم الآلي ، يكون لدى جميع الجماهير ، بما في ذلك أصحاب المصلحة في الأعمال ، سؤال واحد فقط: كيف يعمل النموذج؟ ما هي معايير التقييم الخاصة بالنموذج؟ ما هي دقة النموذج؟ بمعنى آخر ، لتقييم قابلية تعميم النماذج ، لا نحتاج فقط إلى طرق تقدير فعالة وكفؤة ، بل نحتاج أيضًا إلى بعض معايير الأداء التي يمكن أن تحدد قابلية التعميم. تعكس مقاييس الأداء المختلفة المتطلبات المختلفة للقضايا وتنتج نتائج تقييم مختلفة. بمعنى آخر ، جودة النموذج هي مفهوم نسبي يعتمد على الخوارزمية والبيانات وكذلك متطلبات العمل.

يساعدك تقييم النموذج المطور على تعديل النموذج. ومن ثم ، ستستمر في تطوير وتقييم النموذج الخاص بك حتى تصل إلى مستوى كفاءة النموذج الأمثل (لا تعني الكفاءة المثلى للنموذج دقة 100٪ !!). يمكن رؤية العديد من محللي البيانات غير مهتمين بأداء النموذج أو معايير تقييم النموذج. يمكنك إنشاء عدد n من النماذج لمجموعة بيانات معينة ، ولكن السؤال الرئيسي عن النموذج الذي يجب اختياره. ومعايير التقييم النموذجية هي الإجابة على هذا السؤال.

في التنبؤ بالمشكلات ، لدينا مجموعة من البيانات $D = \{(x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)\}$ حيث y_m هي التسمية الحقيقية لعينة x_m. لتقييم أداء المتعلم f ، نقارن توقعه x_m بسمته الحقيقية y .

اعتمادًا على الغرض من عملك ونطاقه ، يمكنك اختيار معايير التقييم النموذجية. عندما نتحدث عن النماذج التنبؤية ، يجب علينا أولاً فهم الأنواع المختلفة للنماذج التنبؤية. بشكل عام، لدينا نوعان من النماذج تعتمدان على المتغيرات التابعة. إذا كان المتغير التابع مستمرًا ، فإننا ننشئ نموذجًا للانحدار ، وعندما يكون المتغير التابع متقطعاً ، يكون لدينا نموذج تصنيف. إذا كانت المشكلة تمثل مشكلة انحدار ، فإن مقياس الكفاءة الأكثر شيوعًا هو متوسط الخطأ التربيعي[1] (MSE):

$$E(f;D) = \frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2$$

[1] Mean Squared Error

في استمرار لهذا القسم ، سوف نقدم وشرح بعض معايير الاداء في قضايا التصنيف.

معدل الخطأ والدقة

مقاييس الأداء الأكثر شيوعًا في مشاكل التصنيف ، بما في ذلك التصنيف الثنائي والتصنيف المتعدد (متعدد الفئات) ، هي معدلات الخطأ[1] والدقة[2]. معدل الخطأ هو نسبة العينات المصنفة بشكل غير صحيح إلى جميع العينات. من ناحية أخرى ، فإن الدقة هي نسبة عينات التصنيف الحقيقية إلى جميع العينات. على سبيل المثال ، إذا كان هناك 100 عينة من x_1 إلى x_{100} والنموذج (المصنف) 20 عينة بشكل صحيح ، فسيكون معدل الخطأ 0. 20/100=0.2. وفقًا لمجموعة البيانات D ، يكون معدل الخطأ على النحو التالي:

$$E(f;D) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(f(x_i) \neq y_i)$$

ونحدد الدقة على النحو التالي:

$$\begin{aligned} Accuracy(f;D) &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(f(x_i) = y_i) \\ &= 1 - E(f;D) \end{aligned}$$

حيث $\mathbb{I}(.)$ هي دالة المؤشر[3] التي ترجع 1 من أجل true و 0 على خلاف ذلك.

الدقة والاستدعاء و *F1*

عندما يتعين علينا تقييم نموذج ما ، فإننا غالبًا ما نستخدم معدلات الخطأ والدقة ، ولكن ما نركز عليه بشكل أساسي هو مدى موثوقية نموذجنا ، وكيف يعمل على مجموعة بيانات مختلفة (قابلية التعميم) ومدى مرونته. لا شك أن الدقة معيار مهم للغاية يجب أخذه في الاعتبار ، ولكنها لا تقدم دائمًا صورة كاملة لأداء النموذج.

عندما نقول أن النموذج موثوق ، فإننا نعني أن النموذج قد حصل على البيانات بشكل صحيح ووفقًا لطلب التعلم. لذلك ، فإن التنبؤات التي قدمتها قريبة من القيم الفعلية. في بعض الحالات، قد يؤدي النموذج إلى دقة أفضل ، لكنه قد لا يفهم البيانات بشكل صحيح وبالتالي يؤدي بشكل

---

[1] Error Rate

[2] Accuracy

[3] indicator function

سيء عندما تكون البيانات مختلفة. هذا يعني أن النموذج ليس موثوقًا وقويًا بدرجة كافية وبالتالي يحد من استخدامه.

على سبيل المثال ، لدينا 980 تفاحة و 20 برتقالة ولدينا نموذج يصنف كل فاكهة على أنها تفاحة. لذلك ، دقة 980/1000=98% ، وبناءً على معيار الدقة لدينا نموذج دقيق للغاية. ومع ذلك ، إذا استخدمنا هذا النموذج للتنبؤ بالثمار المستقبلية ، فسوف نفشل. لأن هذا النموذج يمكن أن يتنبأ بفئة واحدة فقط.

الحصول على صورة كاملة للنموذج ، على سبيل المثال كيف يدرك البيانات وكيف يمكن التنبؤ بها ، يساهم في فهمنا المتعمق للنموذج ويساعد على تحسينه. لذا ، افترض أن لديك نموذجًا يحقق دقة 90٪ ، فكيف يمكنك تحسينه؟ لتصحيح الخطأ ، يجب أن ندركه أولاً. وبالمثل ، لتحسين النموذج ، نحتاج إلى النظر في كيفية عمل النموذج على مستوى أعمق. ومع ذلك ، لا يتم تحقيق ذلك بمجرد النظر إلى معيار الدقة ، وبالتالي يتم النظر في معايير أخرى. معايير مثل الدقة ,والاستدعاء و F1 هي أمثلة على هذه المعايير.

لحساب معايير التقييم لنموذج التصنيف ، نحتاج إلى أربع مجموعات من الفئة الحقيقية وفئة التنبؤ مع العناوين ، الموجب الحقيقي ، الموجب الخاطئ ، المنفي الحقيقي والمنفي الخاطئ، والتي يمكن تمثيلها في مصفوفة الارتباك (الجدول 5-1). أين:

- **موجب حقيقي (*TP*):** على سبيل المثال ، عندما كانت القيمة الفعلية للفئة "نعم" ، توقع النموذج أيضًا "نعم" (أي توقع صحيح).
- **موجب خاطئ (*FP*):** على سبيل المثال ، عندما كانت القيمة الفعلية للفئة "لا" لكن النموذج توقع "نعم" (أي توقع خاطئ).
- **منفى خاطئ (*FN*):** على سبيل المثال ، عندما تكون القيمة الفعلية للفئة "Yes" ، لكن النموذج توقع "لا" (أي توقع خاطئ).
- **منفى حقيقي (*TN*):** على سبيل المثال ، عندما تكون القيمة الفعلية للفئة "لا" وتوقع النموذج "لا" (أي ، كان التوقع صحيحًا).

جدول 5ـ1 مصفوفة الارتباك

		الفئة المتوقعة	
		مثبت	منفى
الفئة الحقيقي	مثبت	موجب حقيقي (TP)	منفى خاطئ (FN)
	منفى	موجب خاطئ (FP)	منفى حقيقي (TN)

الآن ، بناءً على ذلك ، يمكننا تحديد معايير الدقة والاستدعاء و F1:

- **الاستدعاء.** يشير إلى قدرة النموذج على التنبؤ بالحالات الإيجابية من بين جميع الحالات الإيجابية الحقيقية:

$$\text{الاستدعاء} = \frac{TP}{FN + TP}$$

أين يستخدم الاستدعاء؟

يعد الاستدعاء معيارًا جيدًا للحالات التى تكون فيها التكاليف السلبية الخاطئة مرتفعة. على سبيل المثال ، فى تشخيص الاحتيال أو تشخيص المريض يتطلب دخول المستشفى. إذا كان من المتوقع أن تكون المعاملات الاحتيالية غير احتيالية، فقد تكون العواقب وخيمة للغاية على البنك. وبالمثل ، يحتاج المريض إلى دخول المستشفى عند التشخيص. إذا احتاج المريض إلى دخول المستشفى لإجراء اختبار ومن المتوقع أنه لا يحتاج إلى دخول المستشفى وكان المرض معديًا ، فإن تكلفة المنفيات الكاذبة ستكون عالية جدًا.

- **الدقة.** يُظهر جزء الإيجابيات الحقيقية بين العينات التي يُتوقع أن تكون موجبة:

$$\text{الدقة} = \frac{TP}{FP + TP}$$

أين تستخدم الدقة؟

الدقة معيار جيد للحالات التى تكون فيها التكاليف الإيجابية الخاطئة مرتفعة. على سبيل المثال ، اكتشاف البريد الإلكترونى العشوائى. عند اكتشاف البريد الإلكترونى العشوائى ، تعنى النتائج الإيجابية الخاطئة أن البريد الإلكترونى الذى ليس بريدًا عشوائيًا (منفى حقيقى) تم تحديده على أنه بريد عشوائى (بريد عشوائى متوقع). إذا كانت دقة نموذج الكشف عن الرسائل غير المرغوب فيها عالية ، فقد يفقد المستخدم رسائل البريد الإلكترونى المهمة.

- **F1.** يعتبر F1 معيارًا أفضل إذا كنت بحاجة إلى توازن بين الدقة والاستدعاء. ببساطة ، تجمع F1 بين الدقة والاستدعاء في معيار واحد عن طريق حساب الوسط التوافقي بين الاثنين:

$$\text{F1} = 2 \times \frac{\text{الاستدعاء} * \text{الدقة}}{\text{الاستدعاء} + \text{الدقة}}$$

الحساسية[1] و النوعية[2]: ركز على التنبؤات الصحيحة

طبيعة الحساسية والنوعية هي أنهما يركزان على نسبة التنبؤات الصحيحة. لذلك ، يكون النموذج دائمًا هو المعيار للتنبؤات الصحيحة ويكون المقام دائمًا هو إجمالي التنبؤات المقابلة لتلك الفئة. بينما تقيس الحساسية نسبة الإيجابيات الحقيقية المتوقعة إلى جميع القيم الإيجابية الحقيقية ، تقيس النوعية نسبة السلبيات الحقيقية المتوقعة إلى جميع القيم السلبية الحقيقية. تسمى النوعية أيضًا **المعدل السلبي الحقيقي** وتسمى الحساسية **المعدل الإيجابي الحقيقي**.

الحساسية تقاوم السلبيات الكاذبة. الحساسية العالية تعني معدل سلبي كاذب منخفض. بمعنى آخر ، يتم التنبؤ بشكل صحيح بالحساسية لنسبة الإيجابيات الحقيقية:

$$\text{الحساسية} = \frac{TP}{FN + TP} = 1 - \text{المعدل السلبي الكاذب}$$

لاحظ أن معادلات الاستدعاء والحساسية متطابقة رياضياً.

اين تستخدم الحساسية؟
عند تصنيف الإيجابيات ، يكون لها أولوية عالية.
على سبيل المثال: الفحوصات الأمنية في المطارات.

الخصوصية تتعارض مع الإيجابيات الكاذبة. الدقة العالية تعني انخفاض المعدل الإيجابي الكاذب. بمعنى آخر ، فإن الخصوصية هي نسبة السلبيات الحقيقية هي التي يتم توقعها بشكل صحيح:

$$\text{النوعية} = \frac{TN}{TN + FP} = 1 - \text{المعدل الايجابي الكاذب}$$

اين تستخدم النوعية؟
عند تصنيف السلبيات ، يكون لها أولوية عالية.
على سبيل المثال: تشخيص حالة صحية قبل العلاج.

---

[1] Sensitivity

[2] Specificity

طريقة القطع فى نموذج التصنيف

في كل مرة تقوم فيها بتطوير نموذج ، يمنحك هذا النموذج احتمال وقوع الحدث أم لا. يمكنك إجراء تخفيض في الاحتمال الذي تم الحصول عليه. على سبيل المثال ، إذا كان الاحتمال أكبر من 0.50 ، فإن التوقع يكون 1 وبخلاف ذلك يكون التوقع صفرًا. عادةً ما يكون الاحتمال الافتراضي لجميع الخوارزميات في كل برنامج ومكتبة 0.50. ومع ذلك ، يمكنك تغيير نقطة القطع هذه وفقًا لغرض عملك. إذا كنت ترغب في تحمل المزيد من المخاطر ، يمكنك اختيار قطع أقل من 0.50 ، وإذا كنت ترغب في تجنب المخاطرة في توقعك ، يمكنك اختيار قطع أعلى من 0.50.

خلاصة الفصل الخامس

- خوارزمية التعلم الآلي هي إجراء يتم تنفيذه على البيانات لإنشاء "نموذج" للتعلم الآلي.
- نموذج التعلم الآلي "الناتج" هو خوارزمية التعلم الآلي التي تعمل على البيانات.
- لا توجد خوارزمية واحدة تحل جميع مشكلات التعلم الآلي أفضل من أي خوارزمية أخرى.
- إذا كان نموذج التعلم الآلي يعمل بشكل جيد للغاية في بيانات التدريب (خطأ تدريب منخفض) ولكنه يعمل بشكل ضعيف عند اختبار بيانات جديدة (خطأ اختبار مرتفع) ، فهذا عادة ما يكون علامة على أن النموذج يعاني من الضبط الزائد overfitting.
- القدرة على التعلم بقوة هي سبب الضبط الزائد.
- تعد المعاملات الفائقة وسيطات نموذجية يتم تعيين قيمها قبل بدء عملية التعلم.
- يعد ضبط المعاملات الفائقة عملية تحديد التركيبة الصحيحة من المعاملات الفائقة التي تسمح للنموذج بزيادة أداء النموذج إلى الحد الأقصى.

مصادر اضافية إضافية لمزيد من القراءة

Cherkassky, V., & Mulier, F. M. (2007). Learning from data: concepts, theory, and methods. John Wiley & Sons.

Dulhare, U. N., Ahmad, K., & Ahmad, K. A. B. (Eds.). (2020). Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications. John Wiley & Sons.

Golden, R. M. (2020). Statistical Machine Learning: A Unified Framework. Chapman and Hall/CRC.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.

Mello, R. F., & Ponti, M. A. (2018). Machine learning: a practical approach on the statistical learning theory. Springer.

Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

Ott, R. L., & Longnecker, M. T. (2015). An introduction to statistical methods and data analysis. Cengage Learning.

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

Sugiyama, M. (2015). Introduction to statistical machine learning. Morgan Kaufmann.

6

التعلم الخاضع للاشراف

الأهداف:

- ما هو التصنيف؟
- أنواع التصنيف المختلفة.
- الفروق بين أنواع المتعلمين.
- التعرف على خوارزميات التصنيف.
- ما هو الانحدار؟
- أنواع الانحدار.

التصنيف

في التعلم الآلي ، يشير التصنيف إلى مشكلة النمذجة التنبوية التي يتم فيها تعيين كل حالة إلى فئة (أو في بعض الحالات ، مثل التصنيف متعدد العلامات أكثر من واحد). هذه الفئات محددة مسبقًا وغالبًا ما يشار إليها على أنها أهداف أو علامات أو فئات. نظرًا لوجود تسميات في عملية التصنيف ، يقع هذا النهج في فئة التعلم الخاضع للاشراف. يمكن عرض قضايا التصنيف من منظورين مختلفين ، من وجهة نظر عدد العلامات ، والتي يمكن تقسيمها إلى مسألتين: **التصنيف ذو علامة واحدة**[1] **والتصنيف متعدد العلامات**[2] ، ومن منظور عدد الفئات ، يمكن تنقسم إلى مسألتين: **التصنيف الثنائي والمتعدد (متعدد الفئات)**.

التصنيف الثنائي ، حيث يتم تخصيص كل عينة لواحدة فقط من فئتين محددتين مسبقًا ، هو أبسط أنواع التصنيف. يمتد التصنيف الثنائي إلى تصنيفات متعددة من خلال تحديد المزيد من الفئات. التصنيف متعدد العلامات هو شكل عام من التصنيف ذي العلامات الفردية ، حيث يمكن ربط كل حالة بمجموعة من العلامات بدلاً من العلامة.

تعريف التصنيف

التصنيف هو عملية تعيين متغيرات إدخال X جديدة (استناداً إلى نموذج تصنيف يعتمد على بيانات التدريب الموسومة مسبقاً) للفئة الذى ينتمون إليه على الأرجح.

يحاول التصنيف الربط بين الأمثلة التدريبية والفئات المحددة مسبقًا للمشكلة. تُستخدم البيانات المصنفة لتدريب المصنف بحيث تعمل بشكل جيد على بيانات الإدخال الجديدة ويمكنها التنبؤ بالفئة الصحيحة لهذه العينة. بمعنى آخر ، الهدف هو العثور على تقريب جيد لـ $f(x)$ بحيث يمكنه عمل تنبؤات لبيانات لم تظهر فى عملية التدريب وتحديد الفئات التى تنتمى إليها العينة الجديدة..

التصنيف ذو العلامة الواحدة

تقوم المصنفات ذات العلامات الفردية (أو التقليدية) تلقائيًا بتعيين علامة فئة لكل عينة إدخال، حيث يتعلم المصنف أن يرتبط بكل عينة مهملة بالفئة أو الفئة الأكثر احتمالًا. بشكل عام ، يمكن تقسيم مشاكل التصنيف ذو العلامة الواحدة إلى مجموعتين رئيسيتين: المسائل الثنائية والمتعددة.

[1] single-label classification

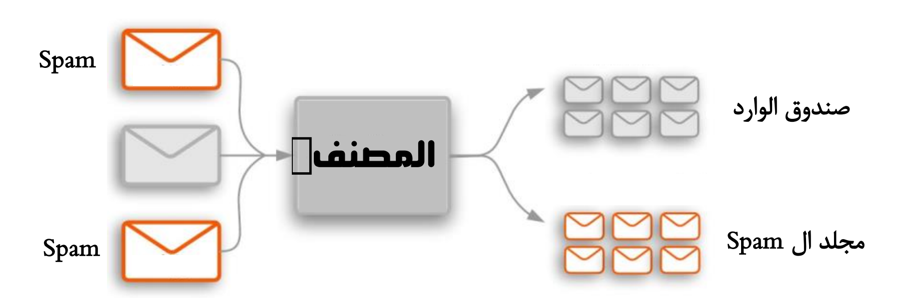
[2] multi-label classification

مشكلة التصنيف الثنائي هي أبسط حالة لمشكلة التصنيف حيث تقتصر مجموعة الفئات على اثنين فقط. في هذا الصدد ، فإننا نميز بين فئة إيجابية وفئة سلبية. مثال بسيط على مشكلة التصنيف الثنائي هو عندما ترى امرأة طبيبًا لتكتشف أنها حامل. قد تكون نتيجة الاختبار إيجابية أو سلبية.

عندما يكون عدد الفئات أكثر من فئتين ، فإن مشكلة التعلم تسمى التصنيف المتعدد. من المفترض أن تكون الفئات المستهدفة منفصلة وفريدة من نوعها. بمعنى آخر ، تنتمي كل حالة إلى فئة واحدة بالضبط. على سبيل المثال ، لدى الشخص فصيلة دم من بين الأنواع الأربعة A أو B أو AB أو O.

التصنيف الثنائي

إنها عملية يتم فيها تصنيف بيانات الإدخال إلى مجموعتين. التصنيف الثنائي هو في الأساس نوع من التنبؤ الذي يتعامل مع أي مجموعة من مجموعتي الفئات تنتمي إليها العينة. افترض أنه تم إرسال رسالتين إلكترونيتين إليك ، واحدة من شركة تأمين ترسل إعلانك والأخرى من البنك بخصوص فاتورة بطاقتك الائتمانية. يصنف موفر خدمة البريد الإلكتروني رسالتي بريد إلكتروني، يتم إرسال البريد الإلكتروني الأول إلى مجلد البريد العشوائي[1] ويتم تخزين البريد الإلكتروني الثاني في البريد الإلكتروني الرئيسي. تُعرف هذه العملية بالتصنيف الثنائي نظرًا لوجود فئتين متميزتين ، أحدهما بريد عشوائي والآخر رئيسي. إذن ، هذه مشكلة تصنيف ثنائي. يوضح الشكل 6ـ1 التصنيف الثنائي.



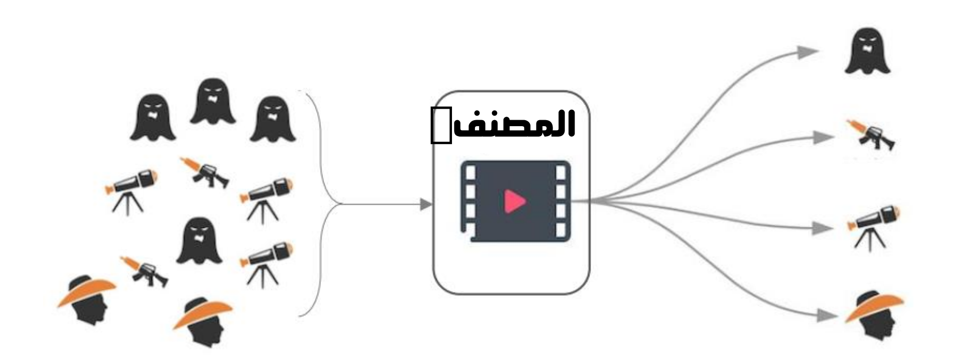
الشكل 6ـ1 التصنيف الثنائي (كشف البريد العشوائي)

التصنيف المتعدد (متعدد الفئات)

التصنيف المتعدد أو متعدد الفئات هو تصنيف العناصر إلى فئات مختلفة. على عكس التصنيف الثنائي ، الذي يقتصر على فئتين فقط ، فإنه لا يوجد حد لعدد الفئات ويمكنه تصنيف أكثر من فئتين. على سبيل المثال ، يعد تصنيف الأخبار إلى فئات مختلفة ، وتصنيف الكتب حسب

[1] Spam

الموضوع ، وتصنيف الحيوانات المختلفة في صورة أمثلة على فئات متعددة (الشكل 2ـ6 هو مثال على فئات متعددة). ومع ذلك ، بينما يتطلب التصنيف الثنائي نموذجًا واحداً فقط للتصنيف ، فإن عدد النماذج المستخدمة في التصنيف المتعدد يعتمد على تقنية التصنيف. في ما يلي ، سوف نصف تقنيتين لخوارزميات التصنيف المتعددة.



الشكل 2ـ6 التصنيفات المتعددة (تصنيف الفيلم حسب الموضوع)

واحد ضد الكل (واحد ضد البقية)[1]

الطريقة الأولى مقابل الكل هي طريقة مبتكرة تستخدمها خوارزمية التصنيف الثنائي لتصنيفات متعددة. تتضمن هذه التقنية تقسيم مجموعة بيانات متعددة الفئات إلى عدة مجموعات من المشكلات الثنائية. في الفئة الأولى مقابل الكل ، لمجموعة البيانات ذات الفئات N ، نحتاج إلى إنشاء فئة N ثنائية. بعد ذلك ، يتم تدريب كل من المصنفات الثنائية لعمل تنبؤ. على سبيل المثال ، على الرغم من وجود مشكلة متعددة التصنيفات مع مجموعات البيانات الحمراء والخضراء والزرقاء ، يمكن إجراء التصنيف الثنائي لحل هذه المشكلة على النحو التالي:

- المسألة 1: **الأحمر مقابل الأخضر / الأزرق.**
- المسألة 2: **الأزرق مقابل الأخضر / الأحمر.**
- المسألة 3: **الأخضر مقابل الأزرق / الأحمر.**

واحد ضد واحد [2]

مثل طريقة واحد للجميع ، فإن طريقة واحد إلى واحد هو طريقة أخرى تستخدم خوارزمية تصنيف ثنائي لتصنيف مجموعات البيانات متعددة الفئات. في هذه الطريقة أيضًا ، يتم تقسيم

---

[1] One vs. All (One-vs-Rest)

[2] One Vs. One

مجموعة البيانات متعددة الفئات إلى مجموعة من التصنيفات الثنائية المتعددة. في التصنيف واحد ضد واحد، لمجموعات البيانات ذات الفئات N ، الرقم:

$$\frac{N * (N - 1)}{2}$$

يتم إنشاء فئة. باستخدام طريقة التصنيف هذا ، تصبح مجموعة البيانات الأصلية مجموعة بيانات، حيث يتناقض كل فئة مع الآخر. على سبيل المثال ، بالنظر إلى مجموعة بيانات متعددة الفئات تحتوي على أربع فئات من الأزرق والأحمر والأخضر والأصفر ، يقسمها طريقة واحد لواحد إلى ست مجموعات بيانات ثنائية.:

- المسألة 1: الأحمر مقابل الأخضر.
- المسألة 2: الأحمر مقابل الأزرق.
- المسألة 3: الأحمر مقابل الأصفر.
- المسألة 4: الأخضر مقابل الأصفر.
- المسألة 5: الأزرق مقابل الأخضر.
- المسألة 6: الأزرق مقابل الأصفر.

التصنيف الفردى

في مشاكل التصنيف الثنائية والمتعددة ، يتم دعم دالة قرار التصنيف من خلال وجود حالات من كل فئة ، وتم تصميم الخوارزميات ذات الصلة لتصنيف كل حالة جديدة في واحدة من عدة فئات محددة مسبقًا. في تطبيقات مثل الأنظمة الصناعية ، تحدد البيانات المتاحة فقط حالات التشغيل العادية للعملية الفيزيائية قيد الدراسة ، بينما يصعب الحصول على البيانات المتعلقة بالحالات غير الفعالة والحالات الحرجة. عندما يتعلق الأمر بالعمليات الصناعية واكتشاف أخطاء الكمبيوتر والتطفل ، بشكل عام ، قد لا يكون عدد أوضاع الفشل والزيادة في عدد الهجمات التي تم إنشاؤها حديثًا محدودًا. هذا هو السبب في أن الباحثين طوروا خوارزميات لحل مشاكل من فئة واحدة على مدى السنوات القليلة الماضية حيث تشير مجموعة البيانات الحالية إلى فئة واحدة.

تتعرف المصنفات الفردية على الأنماط السلوكية العادية للنظام قيد الدراسة. يقومون بتطوير دالات القرار لاختبار عينات جديدة غير متوفرة في قاعدة البيانات التدريبية ، وقبول أكبر عدد ممكن من الحالات وتحديد النقاط البعيدة (أي عينة لا ينتمي إلى نفس توزيع مجموعة البيانات التدريبية).

التصنيف الفردى مقابل التصنيف الثنائى والمتعدد

يختلف التصنيف الفردي عن التصنيف الثنائي والمتعدد لأن المجموعة التدريبية تحتوي فقط على كائنات الفئة المستهدفة ولا تتوفر معلومات حول الفئات الأخرى. وتتمثل مهمتها في وضع الحدود التي تحيط بالأشياء المستهدفة وتقليل فرصة استقبال الأشياء البعيدة.

بمعنى آخر ، يختلف التصنيف الفردي عن التصنيف الثنائي (أو المتعدد) في أنه في الحالة الأخيرة ، تحدد مجموعة التدريب البيانات من جميع الفئات المحددة مسبقًا. ومع ذلك ، فإن مجموعة البيانات الشاملة هذه غير موجودة في تصنيف فئة واحدة. أي أنه لا توجد حالات من الفئة الثانية (أو عدة فئات أخرى) متاحة أثناء التدريب. لذلك ، يتم تصنيف عينات الفئة المتوقعة باستخدام عينات فئة واحدة فقط. بالإضافة إلى ذلك ، نظرًا لأن مشاكل التصنيف الكلاسيكية تخلق دوال تشخيصية بناءً على حالات من جميع الفئات ، فمن الممكن تحديدها بشكل طبيعي. تتطلب هذه الميزة مجموعة بيانات متوازنة لإنشاء حدود قرار فعالة. عندما تكون العينات من فئة ما أكثر وفرة من فئة أخرى ، قد لا تعمل طرق التشخيص بشكل جيد وقد لا يتم استخدامها. نتيجة لذلك ، يصبح التصنيف الفردي ساري المفعول. باختصار ، يوصى بالتصنيف الثنائي (أو متعدد الفئات) إذا كانت مجموعة البيانات تتكون من عدد كبير من عينات جميع الفئات. على العكس من ذلك ، في الحالات غير المتوازنة عند ملاحظة تكرار حالات فئة معينة ، يكون الحل هو استخدام التصنيف الفردي.

في بعض الأحيان ، لا تكون مهمة التصنيف مجرد تعيين عينة اختبار لفئة محددة مسبقًا ، ولكن تحديد ما إذا كانت تنتمي إلى فئة معينة. ومع ذلك ، فإن الغرض من التصنيف التقليدي متعدد الفئات هو تصنيف عينات بيانات غير معروف إلى واحدة من عدة فئات محددة مسبقًا. تنشأ المشكلة عندما لا تنتمي عينة البيانات غير المعروفة إلى أي من هذه الفئات. لنفترض أن لدينا مجموعة بيانات تدريبية تتضمن أمثلة من الفواكه والخضروات. إذا تم تصنيف عينة اختبار غير معروفة (في مجال الفواكه والخضروات ، مثل التفاح أو البطاطس) ، فيمكن استخدام التصنيف الثنائي لهذه المشكلة. الآن إذا كانت عينة البيانات التجريبية من مجال مختلف تمامًا (على سبيل المثال ، قطة من فئة حيوانية) ، فإن المصنف دائمًا يصنف القطة على أنها فاكهة أو خضروات ، وهي نتيجة خاطئة في كلتا الحالتين.

تطبيق التصنيف الفردى

تستخدم التصنيفات الفردية على نطاق واسع في المجالات التالية:

- **في الكشف عن أخطاء الجهاز.** على سبيل المثال ، *عند مراقبة علبة تروس الهليكوبتر، أو في مراقبة حالة تشغيل محطة طاقة نووية ، أو عند اكتشاف انسكاب نفطي. هنا ،* تشمل الفئة السلبية جميع السلوكيات غير الطبيعية المحتملة ، لكنها نادرة جداً في المقام الأول ، وبالتالي ، قد تشكل خطرًا على الناس وتؤدي أيضًا إلى ارتفاع التكاليف.

انتظار حدوث الأخطاء ليس استراتيجية جيدة. بدلاً من ذلك ، يعد بناء التصنيف الفردي بناءً على ملاحظات الآلة العادية حلاً.

- **التشخيص التلقائي للمرض.** يتم تقديم البيانات الإيجابية مع الأمراض "الشائعة" التي يمكن تجميعها بسهولة معًا. بينما تتكون الفئة السلبية من أمراض "نادرة". من الصعب ملء فئة التحكم عن بعد ، لأن اختبارات الأمراض النادرة باهظة الثمن وهناك مرضى نادرون يقللون من فئة العينات السلبية.
- **في المصادقة النشطة للهاتف المحمول.** نحصل فقط على الصور من المستخدم الحالي. نظرًا لصعوبة جمع عينات الفئات السلبية (المستخدمون الآخرون) بسبب مشكلات الخصوصية.
- **برامج كشف الشذوذ[1] في الشبكة.**
- **كشف التسلل في الأنظمة الصناعية.**
- **التعرف على حداثة السلاسل الزمنية.**
- **تحليل نوبات الصرع[2] من إشارات تخطيط كهربية الدماغ داخل الجمجمة.**
- **التعرف على الأشياء المرئية في سياق تفاعلات الإنسان الآلي[3] (HRI).**

يمكن تصنيفها إلى ثلاث تطبيقات عامة: **اكتشاف الحداثة[4]، واكتشاف الشذوذ، والمصادقة الخلوية المتنقلة.**في تشخيص الحداثة، الهدف هو إيجاد حالات جديدة وفقًا للعينات المرصودة. لذلك، من الطبيعي أن تكون بيانات الفئة الجديدة غير معروفة. الغرض من الكشف عن الحالات الشاذة هو تحديد البيانات الشاذة. نظرًا لأن التدريب يتم باستخدام أمثلة القيادة العادية ، يجب أن تتعلم فئتنا مفهوم أن تكون طبيعية..في مصادقة إجراءات الجوال، يتم التحقق من هوية المستخدم باستمرار. تتوفر فقط عيناته للكشف عن العينات السلبية.

تُستخدم خوارزميات التصنيف ذات الفئة الواحدة (التصنيف الاحادى) فى حالة عدم وجود فئة سلبية ، أو ضعف أخذ العينات ، أو عدم تحديدها جيداً.

تصنيف متعدد العلامات

يهتم التعلم الخاضع للإشراف باستنتاج العلاقات بين عينات الإدخال وعلامات الفئة.في مشاكل التصنيف التقليدية، يرتبط كل عينة بعلامة فئة. ومع ذلك،في العديد من سيناريوهات العالم الحقيقي، قد يتم إقران عينة بعلامات متعددة. على سبيل المثال،في فئة الأخبار، يرتبط جزء من

---

[1] anomaly detection

[2] seizure

[3] Human-Robot Interaction

[4] novelty detection

الأخبار حول إطلاق ابل لجهاز الايفون الجديد بكل من العلامة التجارية وعلامة التكنولوجيا. بمعنى آخر، يرتبط كل عينة بمجموعة من العلامات بدلاً من واحدة فقط. التعلم متعدد العلامات هو سياق تعلم آلي يشير إلى التعلم من البيانات متعددة العلامات التي يرتبط فيها كل عينة بالعديد من العلامات المحتملة.

يتمثل الاختلاف الرئيسي بين التعلم متعدد العلامات والتعلم التقليدي الثنائي أو المتعددفي أن العلامات ليست حصريةفي التعلم متعدد العلامات. بمعنى آخر، قد يتم ربط كل حالة بعلامات متعددة. وبالتالي، فإن أحد التحديات الرئيسية للتعلم متعدد العلامات هو كيفية استغلال العلاقات المتبادلة بين العلامات المختلفة بشكل فعال. بالإضافة إلى ذلك، على عكس تصنيف العلامة المفردة، تتأثر مشكلة العلامات المتعددة بالارتباطات المخفية المتأصلة بين العلامات. هذا يعني أن عضوية عينةفي فئة ما يمكن أن تكون مفيدة للتنبؤ بمجموعة العلامات الخاصة بها. على سبيل المثال، يكون الشخص المصاب بارتفاع ضغط الدم أكثر عرضة للإصابة بأمراض القلب أكثر من غيره، ولكن أقل عرضة للإصابة بسوء التغذية العضلي.

التعلم متعدد العلامات

يرتبط التعلم متعدد العلامات بالتنبؤ بعلامات العينات غير المرئية من خلال إنشاء فئة بناءً على البيانات التدريبية. افترض أن $\mathcal{X}$ و $\mathcal{Y}$ يمثلان مساحة عينة الإدخال ومساحة علامة الإخراج، على التوالي.في التعلم متعدد العلامات، يتم تعريف مساحة العلامة $\mathcal{Y}$ على أنها $\mathcal{Y} = \{0,1\}^k$ ، حيث k هو عدد العلامات. أي، إذا كانت العينة مرتبطةً بالتسمية j ، باستثناء أن j هي متجه التسمية 1، وإلا فإن قيمتها تساوي صفرًا. على غرار التصنيف التقليدي، وفقًا لمجموعة بيانات التدريب، فإن الهدف هو تعلم $f(\mathcal{X}) \to \mathcal{Y}$ التعلم متعدد العلامات بواسطة المصنف الذي يتنبأ بعلامات كل عينة $x \in \mathcal{X}$. على وجه التحديد، ناتج الفئة f لعينة معينة هو $x \in \mathcal{X}$:

$$f(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T$$

حيث $f_j(x)(j = 0, \dots, k)$ تساوي واحدًا أو صفرًا، مما يشير إلى علاقة x بالعلامة j.

يمكن أيضًا تعريف التعلم متعدد العلامات بهذه الطريقة. لنفترض أن $\mathcal{X}$ تمثل مساحة عينة الإدخال و $\mathcal{Y} = \{w_1, \dots, w_Q\}$ هي مجموعة محدودة من العلامات. أيضًا ، إذا كانت $\mathfrak{D} = \{(x_1, Y_1), \dots (x_i, Y_i)\}$ تمثل مجموعة البيانات المكونة من n عينات متعددة العلامات $(x_i, Y_i), x_i \in \mathcal{X}, Y_i \subseteq \mathcal{Y}$ ، ثم الغرض من تعلم متعدد العلامات هو إنشاء علامة $\mathcal{H}$ التي تعين عينة x إلى المجموعة المرتبطة من علامات Y وتحسين بعض معايير التقييم.

عندما يتعلق الأمر بالتعلم من البيانات متعددة العلامات ، فهناك طريقتان رئيسيتان لحلها: **تحويل المسألة**[1] **والخوارزمية التكييفية**[2]. في طريقة تحويل المسألة ، يقوم أولاً بتحويل مسألة التعلم متعدد العلامات إلى سلسلة من المسائل أحادية العلامة ثم يحلها باستخدام أساليب التعلم أحادية العلامة الحالية. الهدف من الطريقة الثانية هو تكييف خوارزميات التصنيف الحالية بحيث يمكنها التعامل مع البيانات متعددة العلامات وإنشاء مخرجات متعددة بدلاً من واحدة فقط. بعبارة أخرى ، يعمل تكيف الخوارزمية على توسيع خوارزميات العلامات الفردية للتعامل مباشرة مع البيانات متعددة العلامات.

طريقة تحويل المسألة

أبسط حل للتعلم متعدد العلامات هو طريقة حل المشكلات الذي يمكن استخدامه مع أي خوارزمية تعلم. في هذه الطريقة ، تصبح مشكلة التصنيف متعدد العلامات مشكلة واحدة أو أكثر من مشاكل التصنيف ذات العلامة الواحدة. يتم بعد ذلك دمج حلول هذه المشكلات لحل المشكلة الرئيسية للتعلم متعدد العلامات. تتضمن طريقة حل المشكلات ثلاثة مناهج رئيسية: **الاتصال الثنائي**[3] ، **ومجموعة قوة التسمية**[4] ، **وترتيب العلامة**[5].

الاتصال الثنائي

تقسم طريقة الاتصال الثنائي (BR)، المعروفة أيضًا باسم إستراتيجية الفرد على الكل ، مشكلة العلامات المتعددة مع فئات Q المحتملة إلى مشاكل تصنيف Q ذات علامة واحدة والتي يمكن تعلمها عن طريق تدريس المصنفات الثنائية $Q = (h_1, \dots, h_Q)$. تم حلها. يتم تدريب كل q مصنف $q\epsilon\{1, \dots, Q\}$ على مجموعة البيانات الرئيسية والغرض منه هو تحديد علاقة العلامة الخاصة به بمثيل معين. عند تصنيف عينة جديدة x ، ينتج BR مجموعة من العلامات التي يتم توقعها بشكل إيجابي بواسطة المصنفات الثنائية. ثم يتم تحديد الفئات ذات علامات متعددة بواسطة:

$$\mathcal{H} = \{w_Q \epsilon\, \mathcal{Y} | h_Q(x) = 1\}$$

طريقة BR سهلة التنفيذ وتعقيدها خطي مع عدد العلامات الممكنة. ومع ذلك ، يتجاهل BR الارتباط بين العلامات بكل علامة على حدة. لمواجهة الجوانب السلبية لـ BR ، تم تقديم **سلسلة فئات**[6] **(CC)** ، والتي تتضمن مصنفات ثنائية Q مترابطة على طول السلسلة. طريقة مجموعة

---

[1] problem transformation

[2] adaptation algorithms

[3] binary relevance

[4] label powerset

[5] label ranking

[6] Classifier Chain

قوة العلامة المقدم في القسم التالي هو أيضًا أحد البدائل للتعامل مع هذه الجوانب السلبية لطريقة BR.

مجموعه قوة العلامة

وفقًا لمجموعة التدريب D مع n من العينات ، فإن طريقة مجموعه قوة العلامة (LP) تأخذ في الاعتبار كل مجموعة فريدة من العلامات في D كعلامة ثم يعلم مصنف علامة واحدة. عدد الفئات محدود بالحد الأدنى $\min\left(2^{Q}, n\right)$. يعتمد تعقيد LP على مدى تعقيد الفئات ذات العلامة الواحدة وفقًا لعدد الفئات. للحصول على عينة جديدة ، يُخرج نهج LP الفئة الأكثر احتمالًا ، وهي مجموعة من العلامات في العرض الأصلي متعدد العلامات. تتمتع LP بميزة النظر في ارتباطات العلامة. ومع ذلك ، فإن أحد الجوانب السلبية لهذه الطريقة هي أنه قد تؤدي إلى مجموعة بيانات غير متوازنة مع عدد كبير من الفئات مع عينات قليلة.

طريقة الخوزارزمية التكييفية

تعمل طرق مطابقة المشكلات على تخصيص خوارزميات التعلم الآلي التقليدية لإدارة مفاهيم العلامات المتعددة بشكل مباشر. هذه الأساليب لها ميزة التركيز على خوارزمية محددة. ميزة أخرى هي أن هذه الأساليب تستخدم مجموعة بيانات التدريب بأكملها في وقت واحد لتدريب فئة متعددة العلامات. بشكل عام ، أداء هذه الخوارزميات في مشاكل العالم الحقيقي الصعبة أفضل من طرق حل المشكلات ، وهذا على حساب المزيد من التعقيد.

تطبيقات التصنيف متعددة العلامات

بمجرد تقديم المفاهيم الأساسية المتعلقة بالتصنيف متعدد العلامات، فإن السؤال التالي الذي قد يطرح نفسه هو المكان الذي تنطبق فيه. كما ذكرنا سابقًا، فإن الغرض من المصنف متعدد العلامات هو التنبؤ بمجموعة من العلامات ذات الصلة لعينة بيانات جديدة.في هذا القسم، يتم توضيح العديد من مجالات التطبيق التي يمكن أن تستفيد من هذه الميزة.

تصنيف المشهد[1]

في تصنيف المشهد ، يتم تعيين المهمة لتحديد العلامات الدلالية ذات الصلة مثل الجبل والبحيرة وما إلى ذلك للصور. يتم استخدام تصنيف المشهد في العديد من المجالات ، بما في ذلك فهرسة الصور القائمة على المحتوى وتحسين الصورة الحساسة للمحتوى. على سبيل المثال ، تدعم العديد من أنظمة المكتبات الرقمية الحالية استرجاع الصور المستند إلى المحتوى ، مما يسمح للمستخدم باسترداد الصور التي تشبه صورة الاستعلام. في هذه الحالة ، يمكن أن تؤدي

[1] Scene Classification

معرفة العلامات الدلالية لصورة الاستعلام إلى تقليل مساحة البحث وتحسين دقة الاسترجاع. نظرًا لأن المشهد الطبيعي قد يحتوي على كائنات متعددة ، يمكن ربط كل صورة بعلامات متعددة. ومن ثم ، فإن تصنيف المشهد هو بطبيعة الحال مشكلة تعلم متعددة العلامات.

تصنيف النص

تصنيف النص هو مهمة تصنيف المستندات النصية في مجموعة واحدة أو أكثر من الفئات المحددة مسبقًا. تعود مشكلة تصنيف النص إلى أوائل الستينيات. ومع ذلك ، فقد تحسنت فعالية تصنيف النص بشكل ملحوظ في العقود الأخيرة بسبب التقدم في أساليب التعلم الآلي. يمكن العثور على المستندات النصية في كل مكان ، بدءًا من الشركات الكبيرة التي تخزن مجموعة متنوعة من الاتفاقيات والتقارير إلى الأفراد الذين يقومون بتسجيل فواتيرهم ورسائل البريد الإلكتروني الخاصة بهم. جميع الكتب والمجلات المنشورة وسجلاتنا الطبية التاريخية وكذلك المقالات في الوسائط الإلكترونية ومنشورات المدونات وما إلى ذلك هي أيضًا مستندات نصية.

تم استخدام تصنيف النص في العديد من المجالات مثل تصنيف صفحات الويب والتعرف على موضوع النص وتصفية المحتوى وما إلى ذلك. عادةً ، لا تعتبر العلامات (أو الفئات) المحددة مسبقًا في تصنيف النص حصرية بشكل متبادل. وبالتالي يمكن أن يصنف تصنيف النص بشكل طبيعي على أنه مشكلة تدريبية متعددة العلامات. على سبيل المثال ، ضع في اعتبارك علامات الأعمال والتكنولوجيا والترفيه والسياسة في فئة الأخبار. قد تحتوي مقالة إخبارية حول إطلاق ابل لجهاز الايفون الجديد على علامة تجارية وتقنية تسمية.

التحليل الجينومي الوظيفي[1]

علم الجينوم الوظيفي هو مجال مهم في المعلوماتية الحيوية يدرس وظيفة الجينات والبروتينات من خلال إجراء تحليل واسع النطاق على كميات كبيرة من البيانات التي تم جمعها بواسطة مشاريع الجينوم. على سبيل المثال ، تسمح المصفوفات الدقيقة للحمض النووي للباحثين بقياس مستويات التعبير لآلاف الجينات المختلفة في وقت واحد.

في تحليل التعبير الجيني الآلي ، تتمثل المهمة في التنبؤ بوظيفة الجينات ، وبشكل عام ، تعتمد على افتراض أن الجينات ذات الوظائف المتشابهة لها ملامح تعبير متشابهة في الخلايا. لاحظ أن كل جين قد يرتبط بعدة وظائف في الجينوميات الوظيفية. عندما يتم تسمية الوظائف ، يمكن نمذجة مشكلة التنبؤ الوظيفي في الجينوميات الوظيفية كمشكلة تعلم متعددة العلامات.

---

[1] Functional Genomics Analysis

تحديات التصنيف متعدد العلامات

بالمقارنة مع التصنيفات التقليدية الثنائية والمتعددة ، فإن حل مشكلة التصنيف متعدد العلامات يعد أكثر صعوبة. فيما يلي بعض التحديات الرئيسية في تطبيق التعلم متعدد العلامات بنجاح على قضايا العالم الحقيقي..

يكمن التحدي الأول في كيفية الاستغلال الفعال لهيكل العلامة لتحسين أداء التصنيف. في التعلم متعدد العلامات ، غالبًا ما تكون العلامات مترابطة. لأنها ليست فريدة من نوعها بشكل متبادل. لذلك ، فإن كيفية قياس الارتباطات وتسجيلها في مساحة العلامة مهمة جداً لتحسين التنبؤ.

التحدي الثاني هو فعالية وكفاءة التعلم متعدد العلامات للمشاكل واسعة النطاق ؛ خاصة عندما يكون حجم العلامات وعددها كبيرًا. يعاني التعلم متعدد العلامات أيضًا من **لعنة الأبعاد**[1]، كما أن العديد من طرق التعلم متعددة العلامات الحالية أقل فعالية للبيانات عالية الأبعاد لأن نقاط البيانات مبعثرة ومتباعدة في الفضاء عالي الأبعاد. على سبيل المثال ، أساليب BR و LP التي تمت مناقشتها مسبقًا تقتصر على أحجام علامات صغيرة نسبيًا. في الآونة الأخيرة ، تم اقتراح طرق للتعامل مع عدد كبير من العلامات. على سبيل المثال ، يتم تقليل حجم مساحة العلامة باستخدام تعيين عشوائي. بالإضافة إلى ذلك ، عندما يكون عدد العلامات كبيرًا ، يصبح من الصعب حفظ عدد كبير من نماذج التنبؤ في الذاكرة.

خوارزميات التصنيف

في مشكلة تصنيف التعلم الآلي ، نبدأ بمجموعة بيانات (حيث تأتي النقاط من مساحة العينة) ، جنبًا إلى جنب مع علامة (أو فئة) لكل نقطة (حيث يوجد عدد محدود من العلامات الممكنة). نفترض أن النقاط في مجموعة البيانات يتم توزيعها بشكل مستقل وموحد ، ولدينا نقطة بيانات جديدة من توزيع مشابه لمجموعة البيانات تسمى **الاستعلام**[2] ، ويفترض أيضًا أنها مستقلة عن نقاط مجموعة البيانات. ومع ذلك ، ليس لدينا علامة للاستعلام. وبالتالي ، نريد توقع تسمية الاستعلام بناءً على مجموعة البيانات.

على سبيل المثال ، لنفترض أننا نريد التنبؤ بما إذا كان الشخص معرضًا وراثيًا للإصابة بأمراض القلب. لدينا بيانات من جينومات الأشخاص الذين لديهم تسلسل جيني وما إذا كانوا يعانون من أمراض القلب أم لا. لدينا الآن مريض جديد لدينا جينومه ولكننا لا نعرف ما إذا كان مصابًا بمرض

[1] curse of dimensionality

[2] query

في القلب. لذلك ، نريد أن نتنبأ بما إذا كان الشخص مصابًا بأمراض القلب بناءً على التسلسل الجيني ؛ بالمعلومات الوحيدة المتاحة لنا ، أي مجموعة بيانات الفرد والتسلسل الجيني.

إذا كانت X عبارة عن مجموعة بيانات و Y عبارة عن مجموعة من الفئات ، فإن المصنف $f: X \rightarrow Y$ هو دالة تحاول التنبؤ بالفئة y لنقطة البيانات x. دقة الفئة f هي احتمال توقعنا للعلامة الصحيحة للاستعلام ، وخطأ الفئة f هو احتمال توقعنا للعلامة غير الصحيحة. نريد العثور على مصنف f الذي تكون دقته عالية قدر الإمكان (أو ، على نحو مكافئ ، يكون خطأه صغيرًا قدر الإمكان). عملية تكوين فئة f تسمى **التعلم**. قاعدة التعلم هي مجموعة من الدوال التي تأخذ مجموعة من نقاط البيانات المسماة وتعطي مصنفًا في المخرجات يمكننا استخدامه لتصنيف نقاط الاستعلام. عند تطبيق قاعدة التعلم ثم استخدامها لتصنيف النقاط ، فإننا غالبًا ما نشير إلى مجموعة قاعدة التعلم والفئة كفئة.

لاختبار دقة كل مصنف ، نأخذ مجموعة البيانات ونقسمها إلى مجموعتين فرعيتين منفصلتين، مجموعة التدريب والمجموعة التجريبية. تُستخدم مجموعة التدريب لبناء f ، وبالتالي نتوقع علامات نقاط مجموعة البيانات التجريبية. ثم نقارن العلامات المتوقعة بالعلامات الصحيحة في مجموعة الاختبار ونحسب دقة تنبؤنا.

المتعلمين البارامترية وغير البارامترية

يمكن تصنيف خوارزميات التعلم الآلي إلى فئتين ، بارامتري أو غير بارامتري. يمكن وصف المعامل بأنها متغير تكوين متأصل في النموذج. يمكن اعتبار قيمة المعامل من البيانات التعليمية. بعد التدريب ، يتم استخدام المعاملات لتحديد أداء النموذج في بيانات الاختبار. بمعنى آخر ، يستخدمهم النموذج للتنبؤ. نموذج التعلم الآلي مع عدد من المعاملات هو نموذج بارامتري. باختصار ، عادةً ما تحتوي النماذج البارامترية في التعلم الآلي على نهج قائم على النموذج نقوم فيه بافتراض وفقًا لشكل الدالة المراد تقديرها ثم تحديد النموذج المناسب بناءً على هذا الافتراض لتقدير مجموعة المعاملات. أكبر عيب للطرق البارامترية هو أن الافتراضات التي نتخذها قد لا تكون دائمًا صحيحة. على سبيل المثال ، قد تفترض أن شكل الدالة خطي ، بينما لا يكون كذلك. ومع ذلك ، فإن الطرق البارامترية سريعة جدًا وتتطلب أيضًا بيانات أقل بكثير من الطرق غير البارامترية. من الأمثلة الشائعة على الخوارزمية البارامترية الانحدار الخطي.

في المقابل ، تُعرف الخوارزميات التي ليس لها افتراضات محددة حول نوع دالة التخصيص بالخوارزميات غير البارامترية. نظرًا لعدم وجود افتراض في هذه الطرق ، يمكنهم تقدير الدلة المجهولة f ، والتي يمكن أن تكون بأي شكل. عادةً ما تكون الطرق غير البارامترية أكثر دقة لأنها تسعى للحصول على أفضل ملاءمة مع نقاط البيانات ويمكنها التقاط جوانب أكثر دقة من البيانات. ومع ذلك ، يأتي هذا على حساب الحاجة إلى عدد كبير من الملاحظات لتقدير الدالة المجهولة بدقة f. بالإضافة إلى ذلك ، نظرًا لأن هذه الخوارزميات أكثر مرونة ، فقد تتعلم أحيانًا

الأخطاء والضوضاء بطريقة لا يمكن تعميمها بشكل جيد على نقاط البيانات الجديدة وغير المرئية. باختصار ، يكون التبادل بين المتعلمين البارامتريين وغير البارامتريين في التكلفة الحسابية والدقة. المثال الشائع للخوارزمية غير البارامترية هو خوارزمية كي أقرب جار.

المتعلمين على أساس العينة

المتعلمين المعتمدين على العينة هم خوارزميات تصنيف غير بارامترية تصنف عينة جديدة غير مسماة وفقًا لعينة من التسميات المماثلة في مجموعة التدريب. في جوهر هذه الخوارزميات طريقة بحث بسيطة. يمكن أن تستنتج هذه التقنيات تصنيفات معقدة من عدد صغير نسبيًا من الحالات وتكون مناسبة بشكل طبيعي للمجالات العددية. ومع ذلك ، يمكن أن تكون حساسة للغاية للميزات غير ذات الصلة ولا يمكنها تحديد ميزات مختلفة في مناطق مختلفة من مساحة العينة. بالإضافة إلى ذلك ، على الرغم من أن التعقيد الزمني لتدريب هذه النماذج منخفض ، إلا أن تصنيف عينة جديدة يستغرق وقتًا طويلاً نسبيًا.

أبسط خوارزمية قائمة على العينة هي متعلم **أقرب جار** ، والذي تكون قاعدته لتصنيف نمط غير معروف كما يلي: قم بقياس وتحديد فئة أقرب عينة في مجموعة التعليمات بمعيار مسافة محدد. على الرغم من بساطتها ، تتمتع فئة الجار الأقرب بالعديد من المزايا مقارنة بالطرق الأخرى. على سبيل المثال ، يمكن تعميمها من خلال حزمة تدريبية صغيرة نسبيًا. وهذا يعني ، مقارنة بالطرق الأخرى ، مثل أشجار القرار أو الشبكات العصبية ، أن مصنف أقرب جار يتطلب عينات تدريب أصغر لتحقيق دالة التصنيف نفسها. يمكن لمصنف الجار الاقرب تحقيق دالة يمكنها التنافس مع طرق أكثر حداثة وتعقيدًا مثل أشجار القرار أو الشبكات العصبية.

الفرق بين المتعلمين المستندين إلى العينة والمستندين إلى النموذج

يتم تلخيص الاختلاف الرئيسي بين المتعلمين المعتمدين على العينة والمتعلمين المعتمدين على النموذج في كيفية تعميم معلوماتهم. يحفظ المتعلمون المعتمدون على العينات جميع البيانات الموجودة في مجموعة التدريب ثم يحددون نقطة بيانات جديدة بنفس قيمة المخرجات أو متوسط نقاط البيانات نفسها التي يتم حفظها. من ناحية أخرى ، يقوم المتعلم المستند إلى النموذج بإنشاء خط تنبؤ أو قسم تنبؤ بناءً على الخصائص المختلفة للبيانات التي تم تدريبها. أخيرًا ، يتم وضع نقطة بيانات جديدة على طول هذا الخط أو في أقسام محددة بناءً على خصائصها.

لفهم هذين المتعلمين بشكل أفضل ، يوضح المثال التالي (القصة) الفرق بشكل أفضل. في وسط بلدة صغيرة ، كان هناك متجر ملابس شهير تديره أم وابنتها. يجب أن تعرف الأم مقدار الأموال التي سينفقها العميل في متجرها ، لأنها كانت تكره المفاجآت. قررت ابنتها ، خريجة علوم الكمبيوتر ، بناء نظام حتى لا تضطر والدتها إلى التعامل مع ضغوط عادات الإنفاق غير المعروفة لديها. ينظر النظام إلى خصائص العميل عند دخوله المتجر. تضمنت بعض الميزات

نوع السيارة التي يركبها العملاء وسعر الملابس التي يرتدونها. كان من الشائع في هذه المدينة أن تعكس دائمًا عادات الإنفاق الخاصة بك من خلال سيارتك وملابسك. كان العميل المعتاد للمحل شابًا يدعى شيرفين. كان شيرفين رجل أعمال ناجحًا وأحد أغنى أعضاء المدينة. يقود سيارة BMW ويتباهى باستمرار بملابسه باهظة الثمن. اختار النموذج الذي بنته ابنته عملاء جدد لديهم سيارات فاخرة وملابس باهظة الثمن ، وتوقع أنهم سينفقون ما ينفقه شيرون في المتجر. في معظم الحالات ، كان هذا النموذج ناجحًا تمامًا ، لأن كل شخص ثري في هذه البلدة الصغيرة كان لديه نفس المبلغ تقريبًا من المال. ثم في يوم من الأيام دخل لاعب كرة قدم عظيم اسمه علي المدينة. رفض علي أن يُرى بأي شيء أقل من أحدث طراز لامبورغيني وأفضل الملابس المخصصة التي اشتراها من إيطاليا. عندما اقترب علي من المتجر ، توقع النموذج أن ينفق علي نفس المبلغ الذي ينفقه شيرفين. على الرغم من أن سيارات علي وملابسه كانت أغلى بكثير من شيرفين ، إلا أن هذه كانت أقرب نقطة بيانات يجب أن يشير إليها النظام. أنفق علي في النهاية على المتجر أكثر بكثير من شيرفين. انزعجت الأم من هذا وسمحت لابنتها بإعادة النظر في نموذجها. ومن ثم ، قررت الفتاة استخدام نظام يستخدم المتعلم المستند إلى النموذج بدلاً من النظام الذي تستخدمه (أي المتعلم المستند إلى العينة). وبالتالي ، إذا دخل عميل جديد ، بدون ميزات تعكس بدقة البيانات التي يتذكرها النموذج بالفعل ، فمن المرجح أن تعكس التوقعات المبلغ الذي ينفقه في المتجر. في المرة التالية التي واجهت فيها سيارة غير مألوفة عميلًا يرتدي ملابس غير مألوفة ، قدم النموذج تنبوًا دقيقًا للمبلغ الذي سينفقونه في المتجر. بعد ذلك ، تمكنت الأم من التأكد من أن ذلك لن يحدث مثلما حدث علي.

في هذه القصة ، تمكنا من رؤية موقف لا يوفر فيه نموذج التعلم القائم على العينة تنبوًا دقيقًا. وذلك لأن نقطة البيانات الجديدة (السببية) عملت كنقطة بعيدة مقارنة بالبيانات التي تم تدريب النموذج عليها. يمكن أن تؤدي نماذج التعلم القائمة على العينات أداءً جيدًا للغاية إذا كانت البيانات التي يتم تدريبها مماثلة للبيانات الجديدة التي يحاولون التنبؤ بها. ومع ذلك ، في المواقف التي قد تكون فيها حالات بعيدة ، قد يخطئ النموذج المستند إلى العينة في القيمة المتوقعة. ومع ذلك ، يعتمد نوع النموذج الذي تستخدمه في النهاية لمشكلة التعلم الآلي على الظروف والمواقف.

الفرق بين المتعلمين /الخوارزميات البارامترية وغير البارامترية

المتعلم البارامتري

تضع الأساليب البارامترية افتراضات كبيرة حول تعيين متغيرات الإدخال لمتغيرات الإخراج ، والتي بدورها تكون أسرع في التعلم ، وتتطلب بيانات أقل ، ولكنها قد لا تكون قوية.

أمثلة على هذه الخوارزميات

- الانحدار اللوجستي
- بيربيسترون
- نايف بايز

الفوائد:

- أبسط وأكثر قابلية للفهم ؛ من الأسهل تفسير النتائج.
- التعلم أسرع من البيانات.
- هناك حاجة إلى بيانات تدريبة أقل لتعلم دالة التطبيق.

المحددات:

- قيود النموذج. الأساليب البارامترية تحد الخوارزمية إلى صيغة دالة معينة.
- من غير المحتمل أن يتوافق ضعف الملاءمة في الممارسة مع دالة التطبيق الأساسية. بعبارة أخرى ، لا توفر هذه الأساليب الأنسب للبيانات. هم على الأرجح لا يتطابقون مع دالة التخصيص.
- التعقيد. الخوارزميات البارامترية لها تعقيد محدود. هذا يعني أنها أكثر ملاءمة للمشكلات الأقل تعقيداً.

المتعلم غير البارامتري

تضع الطرق غير البارامتري افتراضات قليلة حول دالة الهدف أو ليس لديها افتراضات حول دالة الهدف ، وتتطلب بدورها بيانات أكثر بكثير ، وتكون أبطأ في التدريب ، ولها تعقيد أعلى للنموذج، ولكنها يمكن أن تنتج نماذج أكثر قوة.

أمثلة على هذه الخوارزميات

- كي–اقرب جار.
- شعاع الدعم الآلي.
- اشجار التصميم مثل CART و C4.5

المزايا:

- قوة عالية من خلال وضع فرضيات ضعيفة أو معدومة حول الدالة الأساسية.
- مرونة عالية ، مما يعني أنها يمكن أن تستوعب عددًا كبيرًا من أشكال الدوال.
- تم إنتاج أداء عالي في النماذج التنبؤية.

العيوب:

- **البيانات التدريبية. هناك حاجة إلى مزيد من بيانات التدريب لتقدير دالة التطبيق.**
- **السرعة. التدريب أبطأ.**
- **الضبط الزائد. بقدر ما تميل هذه الخوارزميات إلى ملاءمة البيانات بشكل أفضل من الخوارزميات البارامترية ، فهي أكثر عرضة للضبط الزائد.**

المتعلم الكسول والمتحمس

عندما تبني خوارزمية التعلم الآلي نموذجًا فور تلقي مجموعة بيانات إرشادية ، يطلق عليها اسم المتعلم المتحمس. يُطلق على هذا النهج الحماسي ، لأنه عندما يتلقى مجموعة البيانات ، فإن أول شيء يفعله هو بناء النموذج. ثم تنسى بيانات التدريب وتستخدم هذا النموذج لاحقًا لتقييمها عندما تأتي بيانات الإدخال. معظم خوارزميات التعلم الآلي مخصصة للمتعلمين المتحمسين.

في المقابل ، عندما لا تبني خوارزمية التعلم الآلي نموذجًا فور تلقي بيانات التدريب ، لكنها تنتظر تقديم بيانات الإدخال للتقييم ، يطلق عليها اسم المتعلم الكسول. تسمى هذه الطريقة بالكسل لأنها تؤخر بناء النموذج حتى يصبح ضروريًا للغاية. بمعنى آخر ، عندما يتلقى بيانات التدريب ، فإنه يخزنها فقط. في وقت لاحق ، عندما تأتي بيانات الإدخال ، عندها فقط تستخدم البيانات المخزنة لتقييم النتيجة. لا يتعلم المتعلم الكسول دالة قابلة للفصل من البيانات التدريبية، ولكنه يتذكر مجموعة البيانات التدريبية. على العكس من ذلك ، يتعلم المتعلم المتحمس وزن نموذجه (المعاملات) بمرور الوقت. من الأمثلة الشائعة على المتعلم الكسول كي_ أقرب جار.

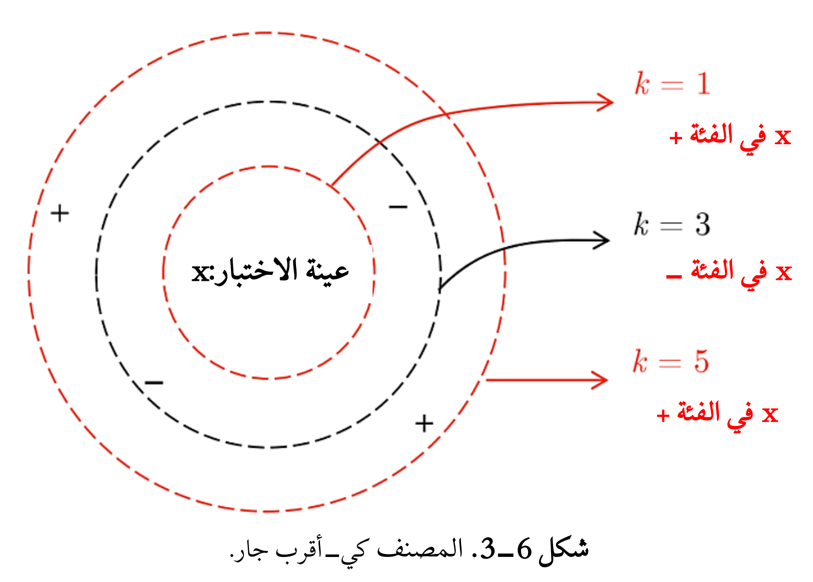
كى-اقرب جار

يعتبر المصنف **كي_اقرب جار (KNN[1])** واحدًا من أقدم وأبسط خوارزميات التعلم الخاضع للإشراف والأكثر فاعلية لتصنيف مجموعات البيانات. تعتمد خوارزمية KNN على افتراض أن الأشياء المتشابهة قريبة من بعضها البعض. بالمقارنة مع خوارزميات التصنيف الأخرى ، فإن أقرب جار يستخدم أسلوب التعلم الكسول. بمعنى آخر ، يقوم ببساطة بتخزين العينات في مرحلة التدريب ولا يفعل شيئًا حتى يتم استلام عينات الاختبار.

يقدم الشكل 6_3 نظرة عامة على فئة الجار الأقرب. كما يتضح ، تلعب المعامل k دورًا مهمًا في تصنيف العينة الجديدة. لأن قيم k المختلفة قد تؤدي إلى نتائج تصنيف مختلفة جدًا. بالإضافة إلى ذلك ، قد تؤدي حسابات المسافة المختلفة إلى جيران مختلفة ، مما يؤدي إلى

[1] The Nearest Neighbor

نتائج تصنيف مختلفة. ومن ثم ، فإن قيمة K المحددة تحدد دقة التنبؤات وعدد الأخطاء ، لذا فإن اختيار K الصحيح له أهمية أساسية في هذه الخوارزمية. يعتمد اختيار K المثالي على البيانات ، لكن الكميات الكبيرة من K تقلل من تأثير الضوضاء على التصنيف ، بينما تقل التمييز بين الحدود والتجمعات.



شكل 6ـ3. المصنف كي_أقرب جار.

يعتمد المصنف كي_اقرب جار على التعلم من خلال **التشابه**[1]. يتم وصف الأمثلة التدريبية بواسطة الخاصية p. تمثل كل عينة نقطة في مساحة p التالية. بهذه الطريقة ، يتم تخزين جميع عينات التدريب في مساحة نمط p. ومن ثم ، عند إعطاء عينة غير معروفه له ، يبحث المصنف كي_الجار في مساحة النمط لعينات تعليمات k الأقرب إلى العينة غير المعروفة. هذه الأمثلة التدريبية k هي أقرب جيران للعينة غير المعروفة. لتحديد أقرب جار لنقطة بيانات ، يجب أن نستخدم معيار التشابه أو الاختلاف بين نقاط البيانات. هناك العديد من معايير التشابه أو الاختلاف ، بما في ذلك المسافة الإقليدية ، ومسافة مينكوفسكي ، ومسافة هيمينج ، ومعامل ارتباط بيرسون ، وتشابه جيب التمام ، وفي هذا القسم ، تم شرح المسافة الإقليدية.

يتم تحديد المسافة الإقليدية على النحو التالي:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{p} (x_{i,l} - x_{j,l})^2}\ , i \neq j.$$

1 analogy

بمعنى آخر ، لكل خاصية عددية ، نأخذ الفرق بين القيم المقابلة لتلك الخاصية في العينة x_i وفي العينة x_j ، نقوم بتربيع هذا الاختلاف ، وفي النهاية يتم أخذ المربع من مجموع عدد المسافات. عادة ، نقوم بتسوية قيم كل خاصية قبل استخدام المعادلة. يساعد هذا في ضمان أن الميزات ذات المجالات الأولية الكبيرة لا تلغي الميزات ذات المجالات الأولية الأصغر.

المسافة الإقليدية هى مقياس للاختلاف بين نقطتى بيانات x_i و x_j. كلما كانت المسافة الإقليدية أكبر ، كلما كانت نقطتا البيانات أكثر اختلافًا.

يتم استخدام خوارزمية KNN لكل من التصنيف والانحدار (التوقع). يحاول KNN التنبؤ بالفئة الصحيحة لبيانات الاختبار عن طريق حساب المسافة بين بيانات الاختبار وجميع نقاط التدريب. عادةً بالنسبة لقضايا التصنيف ، يمكن استخدام التصويت للتنبؤ بعينة الاختبار باعتبارها أكثر تصنيفات الفصل شيوعًا في k الجيران. بالنسبة لمشاكل الانحدار ، يمكن استخدام المتوسط للتنبؤ بالعينة التجريبية على أنها متوسط ناتج k للقيمة الحقيقية.

لا تحتوى خوارزمية كى-اقرب جار على افتراضات حول كيفية توزيع البيانات. وبالتالى، فإن عدم القلق بشأن التوزيع يعد ميزة كبيرة. هذا يعنى أنه يمكن تطبيق KNNs على مجموعات بيانات مختلفة.

كيف تعمل خوارزمية كى- أقرب جار

توضح الخوارزمية التالية كيفية عمل KNN:

خوارزمية كي_ أقرب جار:

الخطوة 1: حدد رقم K المجاور.

الخطوة 2: المسافة الإقليدية (أو معايير المسافة الأخرى) احسب عدد جيران K .

الخطوة 3: رتب المسافة وحدد أقرب الجيران K بناءً على المسافة الإقليدية الدنيا المحسوبة.

الخطوة 4: من هذا الجار K ، احسب عدد نقاط البيانات في كل فئة.

الخطوة 5: قم بتعيين نقاط البيانات الجديدة للفئة مع أقصى عدد من الجيران.

تعتمد خوارزمية KNN على نوع التعلم على النحو التالي:

- **التعلم القائم على العينة:** في هذه الطريقة ، لا نتعلم الأوزان من بيانات التدريب للتنبؤ بالمخرجات (مثل الخوارزميات القائمة على النموذج) ولكننا نستخدم عينات تدريب كاملة للتنبؤ بمخرجات البيانات غير المرئية.
- **التعلم الكسول:** لا يتم تعلم النموذج باستخدام بيانات التدريب السابقة ويتم تأجيل عملية التعلم حتى يتم طلب التنبؤ في العينة الجديدة.

غير البارامترية: في KNN ، لا يوجد شكل محدد مسبقًا لدالة التخصيص.

مميزات خوارزمية كى- أقرب جار:

- إنه سهل التنفيذ.
- وقت التدريب صفر (أو القليل جدًا)
- ليس لديه افتراضات حول كيفية توزيع البيانات.
- من السهل جداً فهم خوارزمية KNN للمبتدئين في التعلم الآلي.

عيوب خوارزمية كى- أقرب جار:

- يجب دائمًا ضبطه على K ، الأمر الذي قد يكون معقداً في بعض الأحيان.
- تكلفة الحسابات عالية بسبب حساب المسافة بين نقاط البيانات لجميع عينات التدريب.
- لا يعمل بشكل جيد على البيانات غير المتوازنة. لذلك ، قد يتم تجميع البيانات الأقل وفرة بشكل غير صحيح.

كى- اقرب جار فى بايثون

في هذا القسم، سنرى كيف يمكن استخدام مكتبة Scikit-Learn بايثون لتنفيذ خوارزمية KNN.

مجموعة البيانات

نستخدم مجموعة iris[1] الشهيرة لمثال KNN الخاص بنا. تحتوي مجموعة البيانات هذه على أربع ميزات: عرض السيبال وطول السيبال وعرض البتلة وطول البتلة. هذه هي خصائص أنواع معينة من نبات السوسن. المهمة هي التنبؤ بالفئة التي تنتمي إليها هذه النباتات. هناك ثلاث فئات في مجموعة البيانات: Iris-setosa و Iris-versicolor و Iris-virginica.

استيراد المكتبات

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

[1] https://archive.ics.uci.edu/ml/datasets/Iris

استيراد مجموعة البيانات

لإدخال مجموعة البيانات وتحميلها بتنسيق بيانات pandas ، قم بتشغيل الكود التالي:

```
In [2]: url = "https://archive.ics.uci.edu/ml/machine-learning-
        databases/iris/iris.data"

        # Assign colum names to the dataset
        names = ['sepal-length', 'sepal-width', 'petal-length',
        'petal-width', 'Class']

        # Read dataset to pandas dataframe
        dataset = pd.read_csv(url, names=names)
```

لمعرفة الشكل الذي تبدو عليه مجموعة البيانات حقًا ، قم بتشغيل الأمر التالي:

```
In [3]:
        dataset.head(8)
```

يؤدي تنفيذ الكود أعلاه إلى عرض الصفوف الثمانية الأولى من مجموعة البيانات كما هو موضح في الصفحة التالية:

	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa

المعالجة الاولية

تتمثل الخطوة التالية في تقسيم مجموعة البيانات إلى سماتها وعلاماتها. استخدم الكود التالي للقيام بذلك:

```
In [4]: X = dataset.iloc[:, :-1].values
        y = dataset.iloc[:, 4].values
```

يحتوي المتغير X على الأعمدة الأربعة الأولى من مجموعة البيانات (أي السمات) بينما يحتوي y على التصنيفات.

تقسيم مجموع البيانات

في الخطوة التالية ، سنقسم مجموعة البيانات الخاصة بنا إلى قسمين ، تدريبي وتجريبي ، مما يمنحنا فكرة أفضل عن كيفية عمل الخوارزمية في المرحلة التجريبية. بهذه الطريقة ، يتم اختبار الخوارزمية الخاصة بنا على بيانات غير مرئية.

لتقسيم البيانات إلى قسمين ، تدريبي وتجريبي ، قم بتشغيل الكود التالي:

```
In [5]: from sklearn.model_selection import train_test_split

        X_train, X_test, y_train, y_test = train_test_split(X, y,
        test_size =0.25, random_state=42)
```

يقسم الكود أعلاه مجموعة البيانات إلى 75٪ بيانات تعليمية و 25٪ بيانات تجريبية. هذا يعني أنه من إجمالي 150 سجلاً ، ستشمل مجموعة التدريب 112 سجلاً وستتضمن مجموعة الاختبار 38 سجلاً:

```
In [6]:  X_train.shape
Out [6]: (112, 4)
In [7]:  X_test.shape
Out [7]: (38, 4)
```

في الكود السابق ، يشير الرقم 4 إلى عدد السمات.

تحجيم الميزة

من الأفضل دائمًا تحجيم الميزات قبل إجراء أي تنبؤات واقعية. الكود التالي يحجم الميزات:

```
In [1]: from sklearn.preprocessing import StandardScaler
        scaler = StandardScaler()
        scaler.fit(X_train)

        X_train = scaler.transform(X_train)
        X_test = scaler.transform(X_test)
```

التدريب والتنبؤ

يعد تعلم خوارزمية KNN والتنبؤ بها أمرًا سهلاً للغاية عند استخدام Scikit-Learn:

```
In [1]: from sklearn.neighbors import KNeighborsClassifier
        classifier = KNeighborsClassifier(n_neighbors=5)
        classifier.fit(X_train, y_train)
```

تتمثل الخطوة الأولى في استيراد كلاس KNeighboursClassifier من مكتبة sklearn.neighbours. في السطر الثاني ، تتم تهيئة هذه الفئة بمعامل n_neigbours ، وهي قيمة K. لقد تم بالفعل ذكر أنه لا توجد قيمة مثالية لـ K ويتم اختيارها بعد الاختبار والتقييم. ومع ذلك ، لتبدأ ، القيم الخمس الأكثر شيوعًا المستخدمة في خوارزمية KNN.

الخطوة الأخيرة هي التنبؤ بالنموذج المبني على بياناتنا التجريبية. للقيام بذلك ، قم بتشغيل التعليمات البرمجية التالية:

```
In [1]:  y_pred = classifier.predict(X_test)
```

تقييم الخوارزمية

لتقييم الخوارزمية ، كما ذكرنا سابقًا ، فإن مصفوفة الارتباك والدقة والاستدعاء ودرجة F1 هي المعايير الأكثر استخدامًا. يمكن استخدام Confusion_matrix و classification_report لحساب هذه المعايير. انظر الكود أدناه:

```
In [3]:  from sklearn.metrics import classification_report,
         confusion_matrix
         print(confusion_matrix(y_test, y_pred))
         print(classification_report(y_test, y_pred))

Out [3]: [[15  0  0]
          [ 0 11  0]
          [ 0  0 12]]
                          precision    recall  f1-score   support

             Iris-setosa       1.00      1.00      1.00        15
         Iris-versicolor       1.00      1.00      1.00        11
          Iris-virginica       1.00      1.00      1.00        12

                accuracy                           1.00        38
               macro avg       1.00      1.00      1.00        38
            weighted avg       1.00      1.00      1.00        38
```

تظهر النتائج أن نموذج KNN الخاص بنا قادر على تصنيف جميع السجلات الـ 38 في مجموعة الاختبار بدقة 100٪ ، وهو أمر ممتاز. على الرغم من أن الخوارزمية عملت جيدًا مع مجموعة البيانات هذه ، فلا تتوقع أن تكون هذه النتائج هي نفسها لجميع مجموعات البيانات !!

مقارنة معدل الخطأ بقيمة K

في قسم التدريب والتنبؤ ، قلنا أنه لا توجد طريقة لمعرفة قيمة K مسبقًا التي ستعمل بشكل أفضل في الخطوة الأولى. اخترنا بشكل عشوائي 5 كقيمة K وصادف أن تكون دقيقة بنسبة 100٪. تتمثل إحدى طرق مساعدتك في العثور على أفضل قيمة لـ K في رسم قيمة K ومعدل الخطأ المرتبط بمجموعة البيانات.

لذلك ، في هذا القسم ، نرسم الخطأ المتوسط للقيم المتوقعة لمجموعة الاختبار لجميع قيم K بين 1 و 30. في هذا الصدد ، نحسب أولاً متوسط الخطأ لجميع القيم المتوقعة التي يختلف فيها K من 1 إلى 30. قم بتشغيل التعليمات البرمجية التالية:

```
In [1]:  error = []

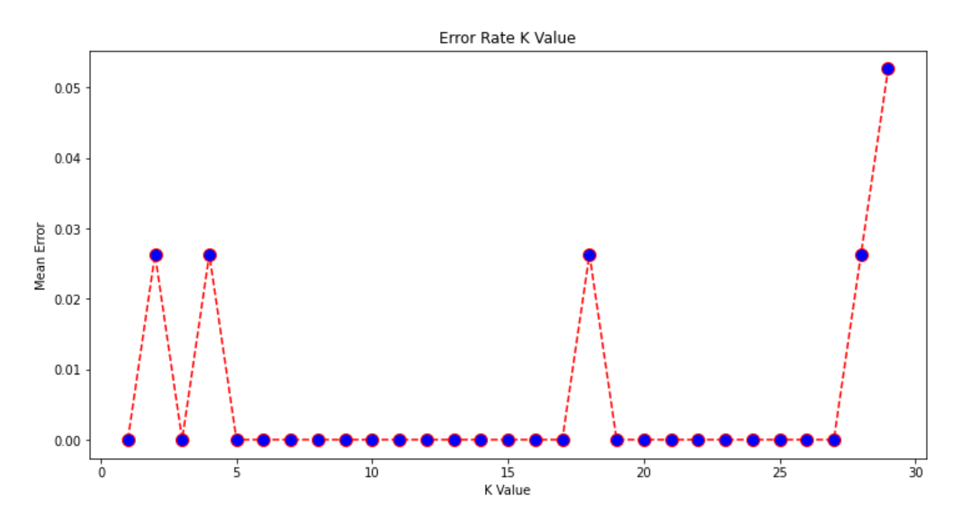
         # Calculating error for K values between 1 and 40
         for i in range(1, 40):
             knn = KNeighborsClassifier(n_neighbors=i)
             knn.fit(X_train, y_train)
             pred_i = knn.predict(X_test)
             error.append(np.mean(pred_i != y_test))
```

يقوم الكود أعلاه بتنفيذ حلقة من 1 إلى 30 وفي كل تكرار يتم حساب متوسط الخطأ للقيم المتوقعة لمجموعة الاختبار وتضاف النتيجة إلى قائمة الأخطاء..

الخطوة التالية هي رسم قيم الخطأ مقابل قيم K. قم بتشغيل التعليمات البرمجية التالية لإنشاء الرسم البياني:

```
In [1]:  plt.figure(figsize=(12, 6))
         plt.plot(range(1, 30), error, color='red',
         linestyle='dashed', marker='o',
                 markerfacecolor='blue', markersize=10)
         plt.title('Error Rate K Value')
         plt.xlabel('K Value')
         plt.ylabel('Mean Error')
```

مخطط الإخراج على النحو التالي:



من خلال هذا الرسم البياني الناتج يمكننا رؤية أفضل وأسوأ قيم K المحددة. يوصى باختبار النموذج بقيم مختلفة لـ K لمعرفة كيف يؤثر على دقة التنبؤات.

شعاع الدعم الآلي

شعاع الدعم الآلي (شبكة المتجهات الداعمة)، أو SVMs[1] باختصار ، هي مجموعة فرعية من طرق التعلم المستخدمة للتصنيف والانحدار واكتشاف النقاط البعيدة. تختلف شبكات المتجهات الداعمة عن خوارزميات التصنيف الأخرى من حيث أنها تحدد حدود القرار التي تزيد المسافة من أقرب نقاط البيانات لجميع الفئات. يسمى حد القرار الذي تم إنشاؤه بواسطة آلات شعاع الدعم الآلي **بالهامش الأقصى أو الهامش الأقصى للصفحة**. يعمل مصنف SVM الخطي البسيط عن طريق إنشاء خط مستقيم (فاصل) بين فئتين. هذا يعني أن جميع نقاط البيانات على جانب واحد من الخط تمثل فئة وأن نقاط البيانات على الجانب الآخر من الخط تمثل فئة مختلفة. من الواضح بشكل بديهي أنه يمكن تحديد عدد لا حصر له من الخطوط. ما يجعل خوارزمية SVM الخطية أفضل من بعض الخوارزميات الأخرى ، مثل كي- أقرب جار، هو أنها تختار أفضل خط لتصنيف نقاط البيانات الخاصة بك.

يساعد المثال ثنائي الأبعاد على فهم هذا بشكل أفضل. افترض أن لديك عدة نقاط بيانات. تحاول فرز نقاط البيانات هذه حسب الفئة التي يجب وضعها فيها ، لكنك لا تريد أن يكون لديك أي بيانات في الفئة الخطأ. هذا يعني أنك تحاول إيجاد خط بين أقرب نقطتين يفصل بين النقاط الأخرى في البيانات. لذا فإن أقرب نقطتي بيانات تعطيك متجهات الدعم التي ستستخدمها للعثور على هذا الخط.

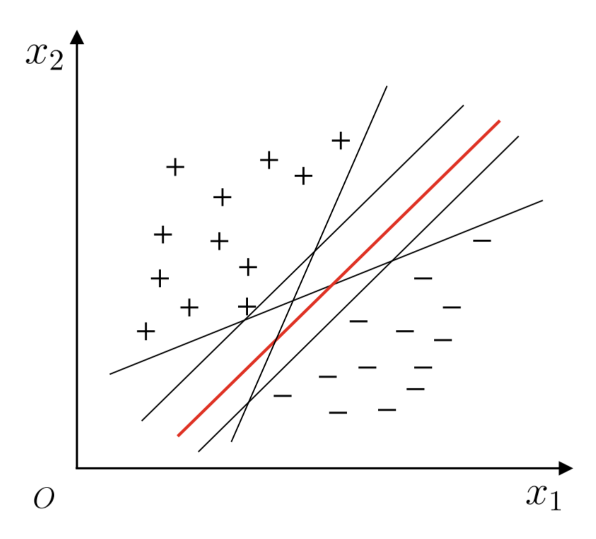
[1] Support-vector machine

النموذج الأساسي والنموذج المزدوج (الثانوي)

يمكن تعريف مشاكل التحسين بطريقتين مختلفتين: المشكلة الأساسية والمشكلة المزدوجة. ميزة هذا هو أنه في بعض الأحيان يكون حل مشكلة مزدوجة أسهل من حل المشكلة الأصلية. ومع ذلك ، قد تكون الحلول للمشكلة الأساسية والمشكلة المزدوجة مختلفة ، ولكن في ظروف معينة ، فإن الحلول متساوية. يمكن أيضًا تعريف شبكة المتجهات الداعمة على أنها مزدوجة واساسية ، نظرًا لأنها مشكلة تحسين. يحصل كلاهما على نفس نتائج التحسين ، لكن تختلف طريقة تلقيهما اختلافًا كبيرًا. قبل أن نتعمق في الرياضيات ، دعني أخبرك أيها يتم استخدامه ومتى. يُفضل النموذج الاساسي عندما لا نحتاج إلى تطبيق خدعة الكيرنل[1] على البيانات وتكون مجموعة البيانات كبيرة ، لكن أبعاد كل نقطة بيانات صغيرة. على العكس من ذلك ، عندما تكون البيانات كبيرة ونحتاج إلى استخدام خدعة النواة ، يُفضل النموذج المزدوج.

شبكة المتجهات الداعمة ذات الهامش الصلب

بالنظر إلى مجموعة التدريب $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$، حيث $y_m \in \{-1, +1\}$الفكرة الرئيسية شعاع الدعم الآلي هي تستخدم مجموعة التدريب D للعثور على مستو فائق فاصل كبير في مساحة العينة يمكنه فصل العينات من الفئات المختلفة. ومع ذلك ، قد يكون هناك عدة مجموعات فاصلة مؤهلة ، كما هو موضح في الشكل 6ـ4. ومن ثم ، أي واحد ينبغي اختياره؟



الشكل 6ـ4. يمكن لأكثر من مستو فائق رئيسي فصل العينات التدريبية.

[1] kernel trick

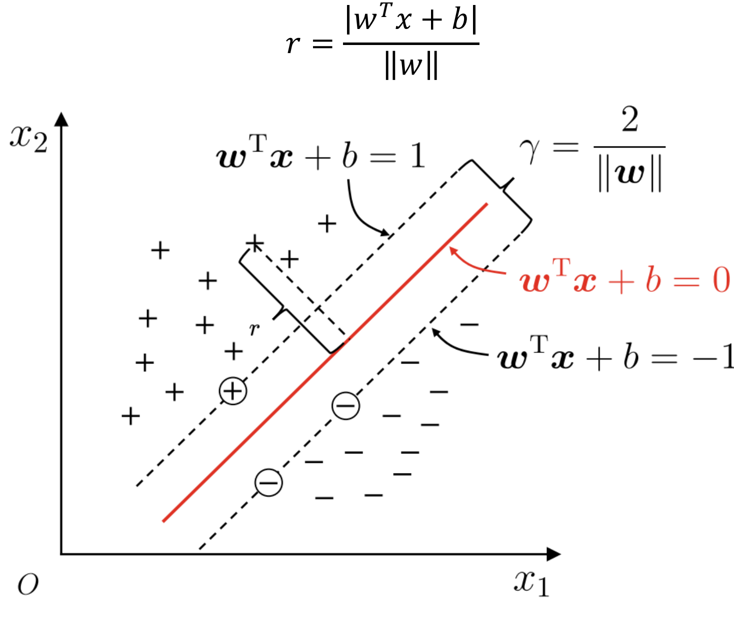
نحتاج إلى تحديد العنصر في منتصف الفئتين ، الأحمر ، من المستو الفائق للفصل (الشكل 6ـ5). لأن هذا الفاصل لديه أفضل "**حد الخطأ المسموح به**[1] " لانحراف البيانات المحلية. على سبيل المثال، قد تكون العينات غير الموجودة في مجموعة التدريب قريبة من حد القرار بسبب الضوضاء أو قيود مجموعة التدريب. نتيجة لذلك ، فإن العديد من المستويات الفائقة للفصل التي تعمل بشكل جيد في مجموعة التدريب سترتكب أخطاء ، بينما تقل احتمالية تأثر الفواصل الحمراء. بعبارة أخرى ، تتمتع هذه الصفحة الفاصلة بأقوى قدرة تعميم في التصنيف.

يمكن تمثيل المستوى الفائق للفصل في فضاء العينة على أنها الدالة الخطية التالية:

$$w^T x + b = 0$$

حيث $w = \{w_1; w_2; \dots; w_d\}$هو متجه طبيعي يتحكم في اتجاه المستو الفائق و b هو الانحياز الذي يتحكم في المسافة بين المستو الفائق والأصل. يحدد المتجه الطبيعي w والتحيز b والمستوى الفائق للفصل ، ويُشار إليه بالرمز w, b. يمكن كتابة المسافة من أي نقطة x في مساحة العينة إلى المستو الفائق w, b على النحو التالي:

$$r = \frac{|w^T x + b|}{\|w\|}$$



الشكل 6ـ5. المتجهات الداعمة والهامش

افترض أن المستو الفائق w, b يمكنه تصنيف الأمثلة التدريبية بشكل صحيح ، أي بالنسبة لـ $(x_i, y_i) \in D$ ، يوجد $w^T x_i + b > 0$ حيث $y_i = +1$ و $w^T x_i + b < 0$ عندما $y_i = -1$ اذا كان لدينا:

[1] tolerance

$$w^T x_i + b \geq +1, y_i = +1,$$
$$w^T x_i + b \leq -1, y_i = -1. \quad (1_6)$$

كما هو مبين في الشكل 6ـ5 ، تم إنشاء المعادلة في (6ـ1) لنقاط العينة القريبة من المستو الفائق. تسمى هذه النقاط متجهات داعمة. مجموع مسافات متجهين داعمين من فئات مختلفة إلى المستو الفائق مساوي

$$\gamma = \frac{2}{||w||}$$

يسمى **الهامش**.

العثور على المستو الفاصل ذات الحد الأقصى للهامش يكافئ إيجاد المعاملات w و b التي تزيد وفقًا للقيود (1ـ6) ، أي:

$$max_{w,b} = \frac{2}{||w||} \quad (2_6)$$

حيث ان $y_i(w^T x_i + b) \geq 1$ ،$i = 1,2, \dots, m$.

يمكن تحسين الهامش عن طريق تكبير $||w||^{-1}$، وهو ما يعادل تصغير $||w||^2$. ومن ثم ، يمكننا إعادة كتابة المعادلة (6ـ2):

$$min_{w,b} \frac{1}{2} ||w||^2 \quad (3_6)$$

حيث ان $y_i(w^T x_i + b) \geq 1$ و $i = 1,2, \dots, m$.

هذا النموذج الاساسي يسمى **آلة متجه الدعم**. يمكننا حل هذه المشكلة بإدخال معاملات لاغرانج α_iوتحويلها إلى مشكلة مزدوجة:

$$L(w, b, \alpha) = \frac{1}{2} w^t w - \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))$$

تسمى دالة لاغرانج هذه بآلة متجه الدعم تكون فيها معاملات b و w للنموذج و $\alpha = (\alpha_1; \alpha_m; \dots; \alpha_m)$مشتق من w و b:

$$\nabla_w L(w, b, \alpha) = 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

$$\nabla_b L(w, b, \alpha) = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y_i = 0$$

استبدالهم في $L(w, b, \alpha)$ يزيل w منه ، ونحصل على مشكلة آلة متجهات الدعم المزدوجة:

$$max_{\alpha}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j\,y_iy_j{x_i}^Tx_j$$

$$s.t.\quad \sum_{i=1}^{m}\alpha_iy_i=0,$$

$$\alpha_i\geq 0, i=1,2,\dots,m.$$

من خلال حل مشكلة التحسين هذه ، يتم الحصول على α وبالتالي w و b. ومن ثم يمكننا كتابة التصنيف النهائي أو دالة القرار على النحو التالي:

$$f(x)=sign[\sum_{i=1}^{m}\alpha_i\,y_i{x_i}^Tx+b]$$

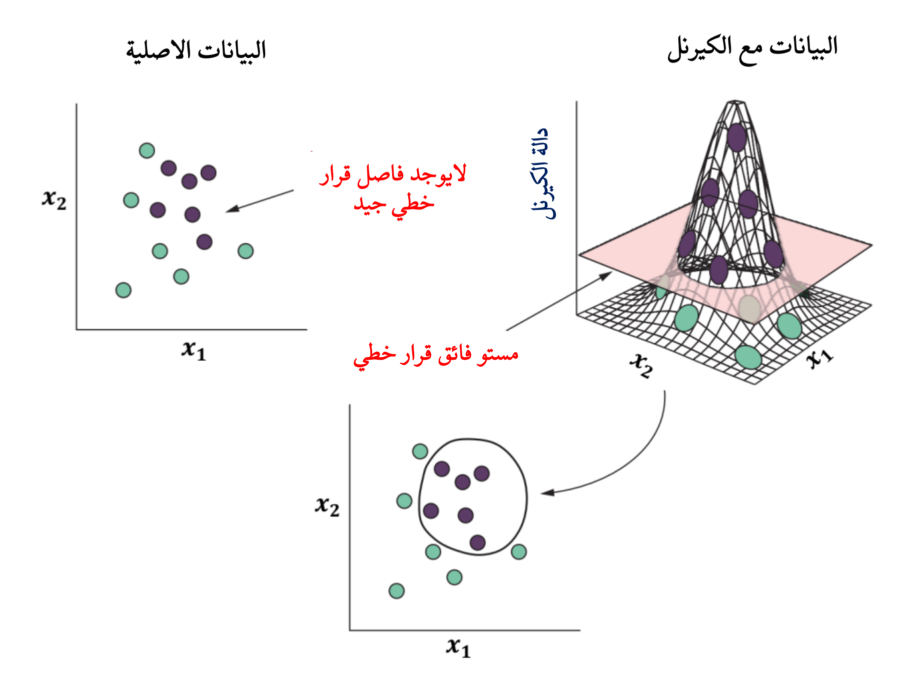
حيث تكون $sign(x)$ **دالة للإشارة**.

حل المشكلات المزدوجة أسهل ، لأنه يحتوي فقط على معامل لاغرانج.

آلة متجه الدعم للبيانات غير الخطية القابلة للفصل

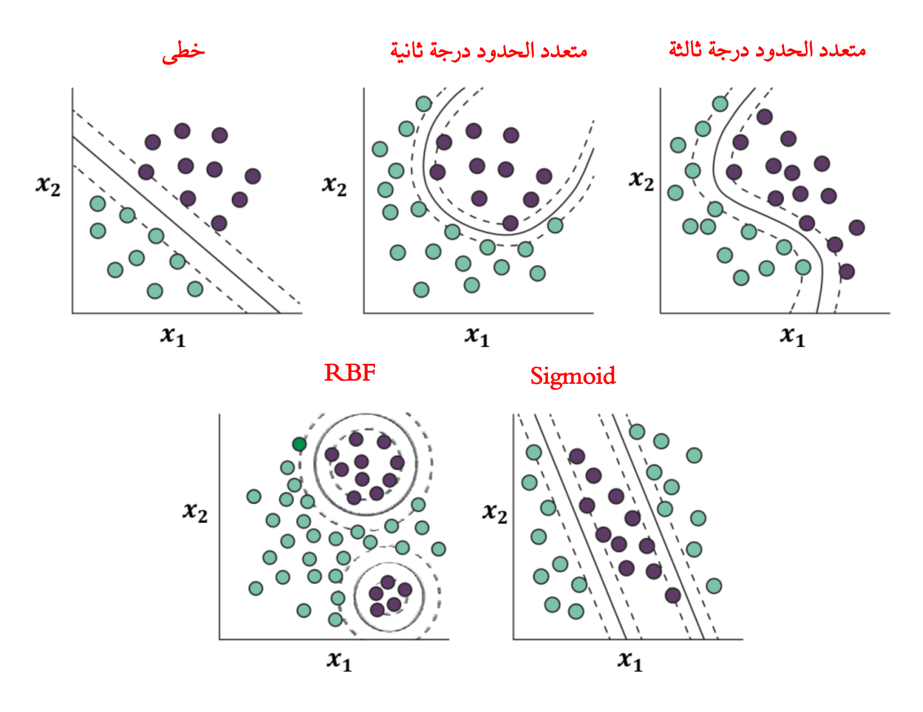
حتى الآن ، يُفترض أن الأمثلة التدريبية قابلة للفصل خطيًا ، مما يعني أن هناك مستويات فائقة يمكنها تصنيف جميع الأمثلة التدريبية بشكل صحيح. ومع ذلك ، فإن هذا الافتراض غالبًا ما يكون غير صحيح في الممارسة. في الواقع ، معظم المشاكل غير خطية ولا يمكن استخدام SVM الخطي السابق لحلها. في هذه الحالة ، ما العمل لحلها؟ من الناحية المثالية ، نحتاج إلى إيجاد تحويل غير خطي φ بحيث يمكن رسم البيانات في مساحة ميزة عالية الأبعاد حيث يكون التصنيف الخطي ممكنًا.

انظر الشكل 6ـ6. لا يمكن فصل الفئات خطيًا باستخدام متغيرين للتنبؤ. تضيف خوارزمية آلة المتجه الداعم بُعدًا إضافيًا للبيانات ، بحيث يمكن للمستو الفائق الخطي فصل الفئات في هذه المساحة الجديدة والأبعاد الأعلى. يمكننا أن نفكر في هذا على أنه نوع من التحول أو امتداد لمساحة الميزة. هذا البعد الإضافي يسمى **الكيرنل**. الكيرنل هي طريقة لحل المشكلات غير الخطية بمساعدة المصنفات الخطية. هذه الفكرة تسمى خدعة الكيرنل**(kernel trick)**.



الشكل 6ـ6. تضيف خوارزمية SVM بُعدًا إضافيًا للفصل الخطي للبيانات. الفئات الموجودة في البيانات الأصلية غير قابلة للفصل خطيًا. تضيف خوارزمية SVM بعدًا إضافيًا ، في مساحة ميزة ثنائية الأبعاد ، يمكن تمثيلها على أنها "تمتد" البيانات إلى البعد الثالث. يسمح هذا البعد الإضافي بفصل البيانات خطيًا.

الآن ، السؤال هو كيف تجد الخوارزمية هذا النواة الجديدة؟ تستخدم الإجابة تحويلًا رياضيًا للبيانات يسمى **دالة الكيرنل**. هناك العديد من الدوال الأساسية للاختيار من بينها ، كل منها يطبق تحويلًا مختلفًا للبيانات ومناسبًا لإيجاد حدود القرار الخطي لمواقف مختلفة. يوضح الشكل 6ـ7 أمثلة على المواقف التي يمكن فيها لبعض دوال الكيرنل الشائعة فصل البيانات غير الخطية القابلة للفصل.



الشكل 6ـ7. أمثلة على دوال الكيرنل

دالة النواة (الكيرنل)

إذا كانت $\varphi(x)$ تمثل الميزة المعينة لـ x ، فيمكن التعبير عن نموذج المستو الفائق للفصل في مساحة الميزة على النحو التالي:

$$f(x) = w^T\varphi(x) + b$$

حيث w و b هما معاملات النموذج. لدينا معادلة مشابهة (6ـ3):

$$min_{w,b}\frac{1}{2}\|w\|^2$$

حيث $y_i(w^T\varphi(x_i) + b) \geq 1$ و $i = 1,2, \dots, m$.

كما أن مشكلتها المزدوجة تساوي:

$$max_\alpha \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j\, y_i y_j \varphi({x_i}^T)\varphi(x_j) \tag{4ـ6}$$

$$s.t. \quad \sum_{i=1}^{m} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0, i = 1,2, \ldots, m.$$

يتضمن حل المعادلة (6ـ4) حساب $\varphi({x_i}^T)\ \varphi(x_j)$، وهو حاصل ضرب الضرب الداخلي لمتجهات الخاصية المعينة x_i و x_j. نظرًا لأن مساحة الميزة المعينة يمكن أن تكون كبيرة جداً أو حتى غير محدودة ، غالبًا ما يكون حساب $\varphi({x_i}^T)\ \varphi(x_j)$ مباشرة أمرًا صعبًا. لتجنب هذه المشكلة ، نفترض وجود الدالة التالية:

$$k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle = \varphi({x_i}^T)\varphi(x_j))$$

والتي تنص على أنه يمكن حساب الضرب الداخلي لـ x_i و x_j في فضاء الميزة في مساحة العينة باستخدام الدالة $k(0,0)$. مع هذه الدالة ، لم نعد بحاجة إلى حساب الضرب الداخلي في مساحة الميزة. ومن ثم يمكننا إعادة كتابة المعادلة (6ـ4) على النحو التالي:

$$max_\alpha \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j\, y_i y_j k(x_i, x_j)$$

$$s.t. \quad \sum_{i=1}^{m} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0, i = 1,2, \ldots, m.$$

سيتعين علينا حل هذه المعادلة:

$$f(x) = w^T \varphi(x) + b$$

$$= sign[\sum_{i=1}^{m} \alpha_i\, y_i \varphi({x_i}^T)\varphi(x) + b]$$

$$= sign[\sum_{i=1}^{m} \alpha_i\, y_i k(x_i, x_j) + b]$$

حيث تكون الدالة $k(0,0)$ هي النواة.

نظرًا لأننا نريد أن تكون العينات قابلة للفصل خطيًا في مساحة الميزة ، فإن جودة مساحة الميزة أمر بالغ الأهمية لأداء آلات متجهات الدعم. ومع ذلك ، لا نعرف أي دوال كيرنل جيدة ، لأننا لا نعرف تعيين السمات. لذلك ، فإن اختيار النواة او الكيرنل هو أكبر عدم يقين لآلات متجهات الدعم. تقوم النواة الضعيفة بتعيين العينات إلى مساحة ميزة ضعيفة ، مما يؤدي إلى ضعف الأداء.

بمعنى آخر ، لا يتم تعلم نوع دالة النواة لمشكلة معينة من البيانات ويجب علينا تحديدها. ومن ثم ، فإن اختيار الدالة الأساسية هو معامل فائق. نتيجة لذلك ، فإن أفضل طريقة لاختيار الدالة الأساسية ذات الأداء الأفضل هي ضبط المعامل الفائق.

انواع دوال الكيرنل

فيما يلي قائمة ببعض الدوال الأساسية المستخدمة في SVM:

- **الكيرنل الخطي.** لنفترض أن لدينا متجهين يسميان x_i و x_j ، يتم تعريف الكيرنل الخطية بواسطة الضرب الداخلي لهذين المتجهين:

$$k(x_i, x_j) = x_i . x_j$$

- **الكيرنل كثيرة الحدود.** يتم تعريف نواة كثير الحدود بالمعادلة التالية:

$$k(x_i, x_j) = (x_i . x_j)^d$$

حيث d هي درجة كثيرة الحدود.

- **كيرنل غاوسي.** معادلة الكيرنل الغاوسية هي كما يلي:

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

يلعب سيجما المعطى دورًا مهمًا جدًا في دوال الكيرنل الغاوسية ويجب تعديله بعناية وفقًا للمشكلة.

- **كيرنل لابلاسيا.** معادلة كيرنل لابلاسيا هي كما يلي:

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{\sigma}\right)$$

- **كيرنل السطح الزائدي او سيكمويد.** تستخدم هذه النواة في الغالب في الشبكات العصبية وتكون معادلتها على النحو التالي:

$$k(x_i, x_j) = \tanh(\alpha x^T y + c)$$

لماذا تكتيك الكيرنل مهم؟

كما ذكرنا سابقًا والموضح في الشكل 6ـ5 ، إذا وجدنا طريقة لتعيين البيانات من الفضاء ثنائي الأبعاد إلى الفضاء ثلاثي الأبعاد ، فيمكننا إيجاد حد قرار يمكنه تصنيف الفئات المختلفة. تتمثل الفكرة الأولى لحل هذه المشكلة حول عملية تحويل البيانات في تعيين جميع نقاط البيانات إلى بُعد أعلى (في هذه الحالة ، 3 أبعاد) ، والعثور على الحدود ، والتصنيف. الفكرة تبدو صحيحة تمامًا. ومع ذلك ، مع زيادة حجم البيانات ، تصبح الحسابات في تلك المساحة أكبر بكثير. هذا هو المكان الذي تكون فيه تكنيك الكيرنل فعالة. لأنه يسمح لنا بالعمل في مساحة الميزة الرئيسية

دون حساب إحداثيات البيانات في مساحة الأبعاد الأعلى. لفهم هذا بشكل أفضل ، دعنا نلقي نظرة على مثال:

$$x = (x_1, x_2, x_3)^T$$
$$y = (y_1, y_2, y_3)^T$$

هنا x و y هما نقطتا بيانات في مساحة ثلاثية الأبعاد. افترض أنه يجب تعيين x و y في فضاء 9 أبعاد. ومن ثم ، يتعين علينا القيام بالحسابات التالية:

$$\varphi(x) = (x_1^2, x_1x_2, x_1x_3, x_2x_1, x_2^2, x_2x_3, x_3x_1, x_3x_2, x_3^2)^T$$
$$\varphi(y) = (y_1^2, y_1y_2, y_1y_3, y_2y_1, y_2^2, y_2y_3, y_3y_1, y_3y_2, y_3^2)^T$$
$$\varphi(x)^T\varphi(y) = \sum_{i,j=1}^{3} x_ix_jy_iy_j$$

للوصول إلى النتيجة النهائية. التعقيد الحسابي ، في هذه الحالة ، هو $O(n^2)$. الآن إذا استخدمنا دالة الكيرنل $K(x, y)$ بدلاً من إجراء حسابات معقدة في مساحة 9 أبعاد ، من خلال حساب الضرب الداخلي لـترانسبوس x و y نحصل على نفس النتيجة في الفضاء ثلاثي الأبعاد مثل التعقيد الحسابي ، في هذا الحالة ، $O(n)$:

$$k(x, y) = (x^Ty)^2$$
$$= (x_1y_1 + x_2y_2 + x_3y_3)^2$$
$$= \sum_{i,j=1}^{3} x_ix_jy_iy_j$$

لفهم هذه المشكلة بشكل أفضل ، ضع في اعتبارك المثال العددي التالي:

إذا كان لدينا نقطتان في الفضاء ثلاثي الأبعاد على النحو التالي:

$$x = (2,3,4)$$
$$y = (3,4,5)$$

أولاً نحسب $\varphi(x)$ ϕ و $\varphi(y)$:

$$\varphi(2, 3, 4) = (4, 6, 8, 6, 9, 12, 8, 12, 16)$$
$$\varphi(3 ,4, 5) = (9, 12, 15, 12, 16, 20, 15, 20, 25)$$
$$\varphi(x). \varphi(y) = \varphi(2, 3, 4). \varphi(3 ,4, 5)$$
$$(36 + 72 + 120 + 72 + 144 + 240 + 120 + 240 + 400) = 1444$$

والحساب $k(x, y)$ يساوي:

$$k(x,y) = (2*3 \ + \ 3*4 \ + \ 4*5)^2$$
$$= (6 \ + \ 12 \ + \ 20)^2$$
$$= 38*38$$
$$= 1444$$

كما رأينا ، كلاهما يعطي نفس النتيجة ، لكن طريقة استخدام الكيرنل تتطلب حسابًا أقل.

توفر النوى اختصارات لتجنب العمليات الحسابية المعقدة.
بشكل أساسي ، ما تفعله **تكتيك الكيرنل** بالنسبة لنا هو توفير طريقة أكثر كفاءة وأقل فعالية من حيث التكلفة لتحويل البيانات إلى أبعاد أعلى. وفقًا لذلك ، لا يقتصر استخدام تكتيكات الكيرنل على خوارزمية SVM. أي عملية حسابية تتضمن الضرب الداخلي (x,y) يمكن أن تستخدم تكتيكات الكيرنل.

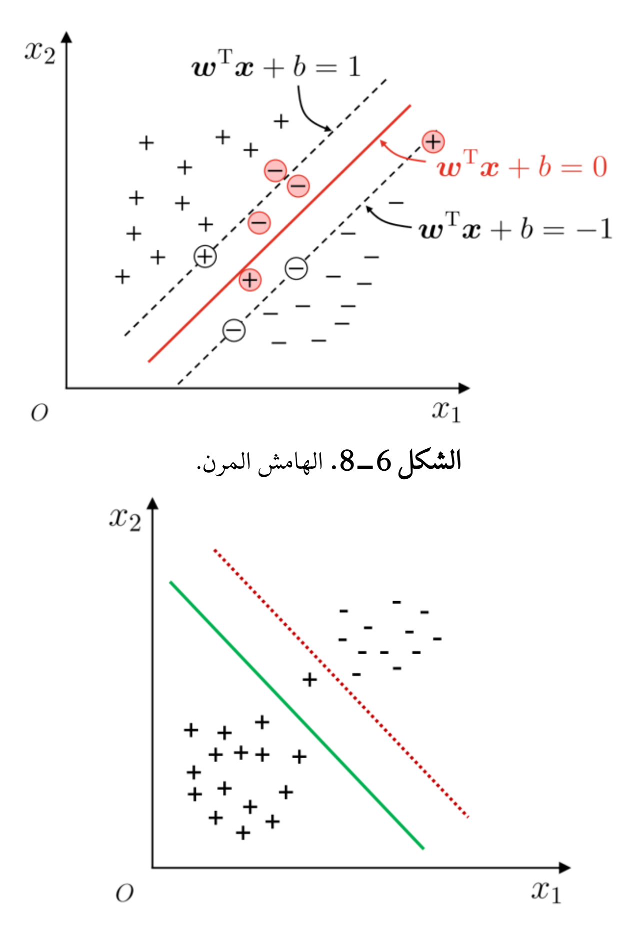
آلة متجهات الدعم ذات الهامش المرن

حتى الآن ، افترضنا أن العينات قابلة للفصل خطيًا في مساحة العينة أو مساحة الميزة. ومع ذلك، غالبًا ما يكون من الصعب العثور على دالة أساسية مناسبة للفصل الخطي لعينات التدريب في مساحة الميزة. يمكن أن يُعزى ذلك إلى حقيقة أن الخصائص التي نحصل عليها من البيانات لا تحتوي عادةً على معلومات كافية للتمكن من فصل الفئات بوضوح (هذا هو الحال عادةً في العديد من تطبيقات العالم الحقيقي). حتى لو وجدنا مثل هذه الدالة الأساسية ، فمن الصعب معرفة ما إذا كانت هذه النتيجة overfitting.

تتمثل إحدى طرق تقليل هذه الحالة في السماح لـ SVM بارتكاب عدد معين من الأخطاء في العينات بحيث تظل النقاط الأخرى مصنفة بشكل صحيح. يتم تنفيذ هذه الفكرة بمفهوم **الهامش المرن**، يوضح الشكل 6ـ8 هذه الفكرة. باختصار ، الدافع لاستخدام هذه الطريقة لسببين:

1. كما ذكرنا سابقًا ، تحتوي معظم تطبيقات العالم الحقيقي على بيانات لا يمكن فصلها خطيًا.
2. أيضًا ، في حالات نادرة حيث تكون البيانات قابلة للفصل خطيًا ، قد لا نرغب في اختيار حد القرار الذي يفصل البيانات تمامًا. بمعنى آخر ، نريد تجنب Overfitting. خذ الشكل 6ـ9 ، على سبيل المثال. هنا تفصل حدود القرار الحمراء تمامًا بين جميع النقاط التدريبية. ومع ذلك ، هل من الجيد حقًا أن يكون لديك حدود قرار بهامش منخفض؟ هل تعتقد أن حدود مثل هذا القرار تمتد بشكل جيد إلى البيانات غير المرئية؟ الجواب لا. يكون لحد القرار الأخضر هامش أوسع

يسمح له بالتعميم بشكل أفضل على البيانات غير المرئية. وبالتالي ، فإن آلة المتجهات الداعمة ذات الحافة المرنة تساعد على تجنب مشكلة Overfitting.



الشكل 6ـ8. الهامش المرن.

الشكل 6ـ9. أفضل حدود القرار؟ أخضر أو أحمر (متقطع)؟

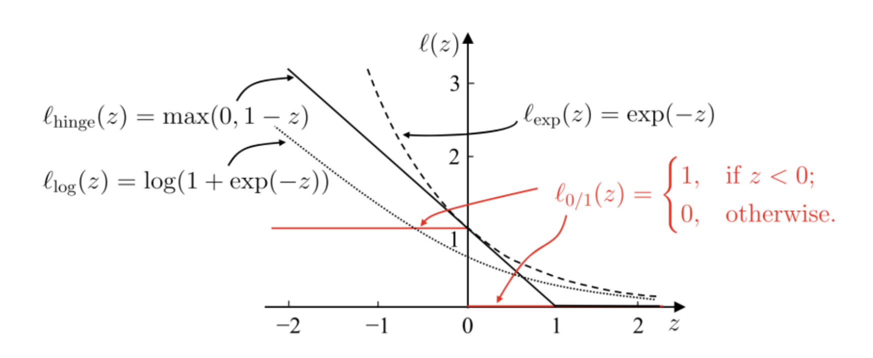
بتعبير أدق ، فإن SVM (المعادلة 6ـ1) الذي تم تقديمه مسبقًا يخضع لقيود ، مما يعني أن الهامش الصلب يجب أن يصنف جميع العينات بشكل صحيح وبدون أخطاء. ومع ذلك ، فإن الهامش المرن يسمح بانتهاك هذا التقييد (عن طريق ارتكاب خطأ في حالات متعددة). بالطبع، يجب أن يقلل الهامش المرن من عدد الحالات التي تنتهك القيد ، مع زيادة الهامش إلى الحد الأقصى. ومن ثم ، يمكن كتابة هدف التحسين على النحو التالي:

$$min_{w,b}\, \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \ell_{\frac{0}{1}}(y_i(w^T x_i + b) - 1) \tag{5ـ6}$$

حيث $C > 0$ ثابت و $\ell_{0/1}$ دالة الخطأ قدرها 0.1. هنا ، C هو معامل فائق تحدد المقايضة بين تعظيم الهوامش وتقليل الأخطاء. عندما تكون صغيرة جداً ، يتم إعطاء أهمية أقل لأخطاء التصنيف ويكون التركيز أكثر على تعظيم الهوامش. بينما عندما تكون كبيرة بشكل لا نهائي ، يكون التركيز أكثر على تجنب التشخيص الخاطئ على حساب الحفاظ على الهامش صغيرًا. بمعنى آخر ، عندما تكون كبيرة بشكل لا نهائي ، فإنها تجبر جميع العينات على اتباع القيد ، وهو ما يعادل آلة متجه داعمة ذات هامش صلب (المعادلة 6ـ3).

يصعب حل المعادلة (6ـ5) مباشرة. لأن 0.1 لها خصائص رياضية ضعيفة ، أي أنها غير محدبة ومتقطعة. لذلك ، غالبًا ما نستبدل 0.1 ببعض دوال الخطأ الأخرى التي لها خصائص رياضية جيدة (على سبيل المثال ، محدب ومستمر). يوضح الشكل 6ـ10 ثلاث دوال خطأ شائعة:

- **خطأ هينج:** $\ell_{hinge}(z) = max(0, 1 - z)$
- **خطأ أسي:** $\ell_{exp}(z) = exp(-z)$
- **خطأ لوجستي:** $\ell_{log}(z) = log(1 + \exp(-z))$



الشكل 6ـ10. دوال الخطأ البديلة لـ $\ell_{0/1}$

عند استخدام دالة خسارة هينج، يتم تحويل المعادلة 6ـ5 على النحو التالي:

$$min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} max(0,1 - y_i(w^T x_i + b)) \qquad (6\text{ـ}6)$$

بإدخال المتغير المساعد $\xi_i \geq 0$ ، تتم إعادة كتابة المعادلة 6ـ6 على النحو التالي:

$$min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i$$

بهذه الطريقة $y_i(w^T\varphi(x_i)+b) \geq 1-\xi_i$، $\xi_i \geq 0$ و $i=1,2,\ldots,m$. تُستخدم هذه المعادلة بشكل شائع لألات المتجهات الداعمة ذات الهامش المرن. في هذه المعادلة ، يحتوي كل عينة على متغير مساعد مقابل يشير إلى درجة انتهاك القيد.

تتحكم عبارة C في دالة الهدف في التوازن بين تعظيم الهوامش والتأكد من أن هوامش الأداء عالية قدر الإمكان. يمنحنا هذا تأثيرًا تنظيميًا يخبر SVM بعدم overfitting (لا تتلاءم تمامًا مع البيانات) حتى إذا كانت هناك قيم بعيدة في مجموعة البيانات.

الهوامش الصلبة ام المرنة؟

يمكن استخدام الهوامش الصلبة أو المرنة في آلة المتجهات الداعمة في دقة البيانات. إذا كانت بياناتنا قابلة للفصل خطيًا ، فإننا نذهب إلى الحافة الصلبة. ومع ذلك ، في ظل وجود نقاط البيانات التي تجعل من المستحيل العثور على التصنيف الخطي ، نحتاج إلى أن نكون أكثر تساهلاً ونسمح بتصنيف بعض نقاط البيانات بشكل غير صحيح. بمعنى آخر ، نستخدم هوامش مرنة.

في بعض الأحيان ، تكون البيانات قابلة للفصل خطيًا ، لكن الهامش صغير جداً بحيث يكون النموذج عرضة للضبط الزائد الحساسية المفرطة للعناصر البعيدة. ومن ثم ، في هذه الحالة ، للمساعدة في تصميم النموذج بشكل أفضل ، يمكننا تحديد هامش أكبر باستخدام SVM ذات الهامش المرن.

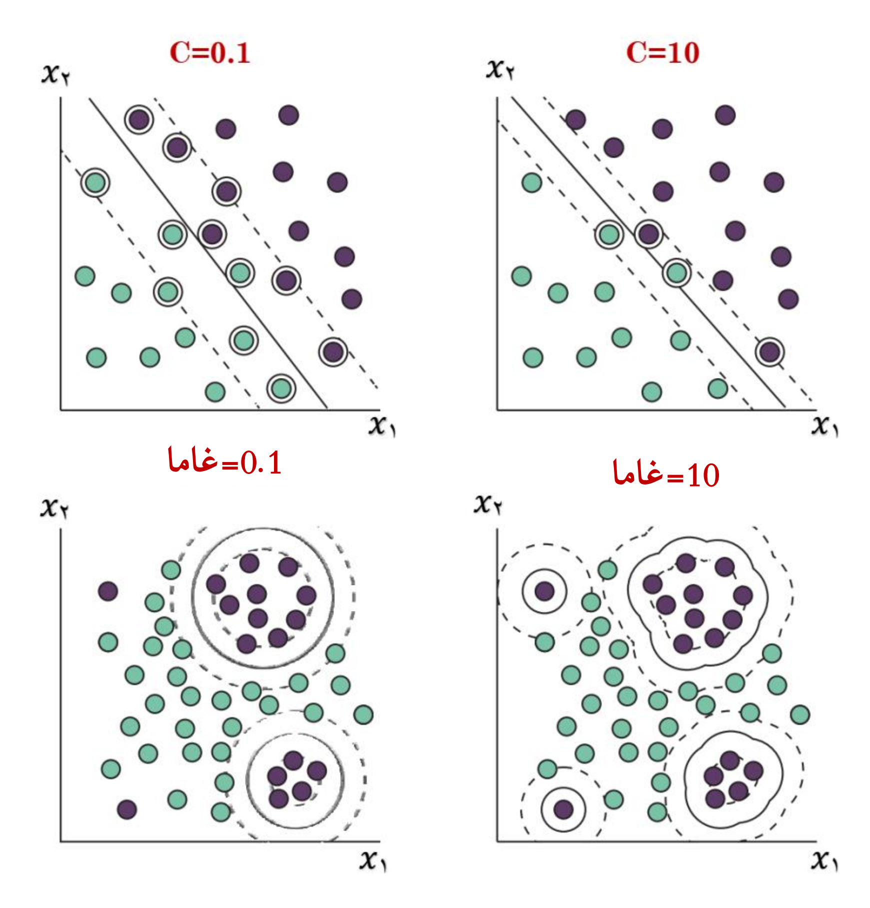
وتجدر الإشارة إلى أن مشاكل العالم الحقيقي غالبًا لا يمكن فصلها خطيًا ، لذلك لا يمكنك استخدام الهوامش الصلبة في هذه المشكلات. ومع ذلك ، إذا وجدت مخططًا لنواة يفصل البيانات المحولة خطيًا ، يمكنك استخدام الهوامش الصلبة.

معاملات آلة المتجهات الداعمة

قبل تدريب النماذج ، علينا ضبط المعاملات الفائقة. تعد المعاملات الفائقة ضرورية لبناء نماذج قوية ودقيقة. إنها تساعدنا في إيجاد التوازن بين التحيز والتباين وبالتالي تمنع النموذج من الضبط الزائد أو الناقص. عند إنشاء نموذج قائم على SVM ، نحتاج أيضًا إلى تكوين كميات كبيرة من المعاملات الفائقة ، والتي يتم سرد أهمها أدناه.:

- **معامل الكيرنل** (شكل 6ـ7).
- **معامل الدرجة** الذي يتحكم في مدى مرونة حدود القرار للنواة متعددة الحدود (الشكل 6ـ7). كلما زادت درجة تعددية الحدود ، أصبحت حدود اتخاذ القرار أكثر مرونة وتعقيدًا. ومع ذلك ، من المحتمل أن يتسبب في زيادة overfitting.

- **معامل التكلفة أو** C الذي يتحكم في كيفية الهامش "الهامش" أو "الناعم" (الشكل 6ـ11).
- يتحكم **معامل غاما** في تأثير نقطة التدريب على موضع حدود القرار. يتم استخدام هذه المعامل الفائقة من خلال الدالة الأساسية للقاعدة الشعاعية. تشير قيم غاما المنخفضة إلى نصف قطر تشابه مرتفع ، مما يؤدي إلى المزيد من تجميع النقاط. بالنسبة لقيم غاما المرتفعة ، يجب أن تكون النقاط قريبة جداً من بعضها حتى يتم أخذها في الاعتبار في مجموعة (أو فئة). لذلك ، تميل النماذج ذات قيم غاما الكبيرة جداً إلى overfitting. كلما كانت غاما أصغر ، قل الاهتمام الذي سيتم توجيهه لكل حالة وكلما انخفضت حدود اتخاذ القرار (من المحتمل أن يؤدي ذلك إلى underfitting). يظهر تأثير غاما لنواة القاعدة الشعاعية الغاوسية في الجزء السفلي من الشكل 6ـ11.



الشكل 6ـ11. تاثير معاملات التكلفة وغاما.

معامل غاما ضد معامل C

بالنسبة للنواة الخطية ، نحتاج فقط إلى تحسين المعلمة C. ومع ذلك ، إذا أردنا استخدام نواة شعاعية ، فيجب تحسين كل من المعامل C وغاما في وقت واحد. إذا كانت غاما كبيرة ، فإن تأثير C يكون ضئيلًا. إذا كانت غاما صغيرة ، فإن C تؤثر على النموذج ؛ بالضبط كما يؤثر على النموذج الخطي. القيم النموذجية لـ C وغاما هي كما يلي. ومع ذلك ، قد توجد بعض القيم المثلى اعتمادًا على التطبيق:

$$0.0001 < \text{غاما} < 10$$

$$0.01 < \mathrm{C} < 100$$

مميزات آلة المتجهات الداعمة:

- إنه فعال للغاية حتى مع البيانات الكبيرة.
- يمكن إدارة البيانات غير الخطية بشكل فعال باستخدام تكتيكات النواة.
- يمكن استخدامها لحل مشاكل التصنيف والانحدار.

عيوب آلة المتجهات الداعمة:

- علينا أن نختار نواة مثالية لـ SVM ، وهو أمر صعب.
- في مجموعة البيانات الكبيرة ، يستغرق التدريب وقتًا أطول نسبيًا.
- آلة المتجه الداعمة ليست نموذجًا احتماليًا ، لذلك لا يمكننا شرح التصنيف من حيث الاحتمالية.
- نموذج SVM (مقارنة بشجرة القرار) يصعب فهمه وتفسيره.

آلة المتجه الداعم في بايثون

مجموعة البيانات

بالنسبة إلى SVM ، على سبيل المثال ، نستخدم مجموعة بيانات iris كما في عينة KNN.

استيراد المكتبات

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

استيراد مجموعة البيانات

```
In [2]: url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

        # Assign colum names to the dataset
        names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']

        # Read dataset to pandas dataframe
        dataset = pd.read_csv(url, names=names)
```

لعرض صفوف وأعمدة مجموعة البيانات ، قم بتنفيذ الأمر التالي:

```
In [3]: dataset.shape

Out [3]: (150, 5)
```

في الإخراج ترى القيمة (150.5) التي تشير إلى أن مجموعة البيانات بها 150 عينة مع 5 أعمدة.

المعالجة المسبقة

تتمثل الخطوة التالية في تقسيم مجموعة البيانات الخاصة بنا إلى سماتها وعلاماتها. استخدم الكود التالي للقيام بذلك:

```
In [4]: X = dataset.iloc[:, :-1].values
        y = dataset.iloc[:, 4].values
```

يحتوي المتغير X على الأعمدة الأربعة الأولى من مجموعة البيانات (أي السمات) بينما يحتوي y على التصنيفات (العلامات).

تقسيم مجموعه البيانات

في الخطوة التالية ، سنقسم مجموعة البيانات الخاصة بنا إلى قسمين ، تدريبية وتجريبية ، مما يمنحنا فكرة أفضل عن كيفية عمل الخوارزمية في المرحلة التجريبية. بهذه الطريقة ، يتم اختبار الخوارزمية الخاصة بنا على بيانات غير مرئية.

لتقسيم البيانات إلى قسمين ، تدريبية وتجريبية ، قم بتنفيذ الكود التالي:

```
In [5]: from sklearn.model_selection import train_test_split

        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =0.25, random_state=42)
```

يقسم الكود أعلاه مجموعة البيانات إلى 75٪ بيانات تدريبية و 25٪ بيانات تجريبية. هذا يعني أنه من إجمالي 150 سجلاً ، ستشمل مجموعة التدريب 112 سجلاً وستتضمن مجموعة الاختبار 38 سجلاً.:

```
In [6]:  X_train.shape
Out [6]: (112, 4)
In [7]:  X_test.shape
Out [7]: (38, 4)
```

في الكود السابق ، يشير الرقم 4 إلى عدد السمات.

تحجيم الميزة

من الأفضل دائمًا تحجيم الميزات قبل إجراء أي تنبؤات واقعية. الكود التالي يحجم الميزات:

```
In [1]:  from sklearn.preprocessing import StandardScaler
         scaler = StandardScaler()
         scaler.fit(X_train)

         X_train = scaler.transform(X_train)
         X_test = scaler.transform(X_test)
```

التدريب والتنبؤ

أولاً قمنا بتقسيم البيانات إلى مجموعات تدريبية وتجريبية ثم قمنا بتحجيم الخصائص على البيانات. حان الوقت الآن لتدريب SVM على بيانات التدريب. يتضمن Scikit-Learn مكتبة svm تتضمن فئات مضمنة لخوارزميات SVM المختلفة. نظرًا لأننا سنقوم بمهمة تصنيف ، فإننا نستخدم فئة مصنف المتجه الداعم ، والذي تمت كتابته كـ SVC في مكتبة Scikit-Learn. تحتوي هذه الفئة على معامل من نوع kernel. في ما يلي ، سوف نستخدم ثلاثة أنواع من النوى: خطي ، كاوسيان و Sigmoid. في طريقة fit ، يتم استدعاء كلاس SVC لتعليم الخوارزمية على بيانات التدريب. قم بتنفيذ الكود التالي لتعليم الخوارزمية بنواة خطية:

```
In [1]:  from sklearn.svm import SVC
         classifier = SVC(kernel='linear')
         classifier.fit(X_train, y_train)
```

الخطوة الأخيرة هي التنبؤ بالنموذج المبني على بياناتنا التجريبية. للقيام بذلك ، قم بتنفيذ التعليمات البرمجية التالية:

```
In [1]:  y_pred = classifier.predict(X_test)
```

تقييم الخوارزمية

```
In  [3]:  from sklearn.metrics import classification_report,
          confusion_matrix
          print(confusion_matrix(y_test, y_pred))
          print(classification_report(y_test, y_pred))

Out [7]:  [[15  0  0]
           [ 0 10  1]
           [ 0  0 12]]
                          precision    recall  f1-score   support

             Iris-setosa       1.00      1.00      1.00        15
         Iris-versicolor       1.00      0.91      0.95        11
          Iris-virginica       0.92      1.00      0.96        12

                accuracy                           0.97        38
               macro avg       0.97      0.97      0.97        38
            weighted avg       0.98      0.97      0.97        38
```

التدريب بالنواة غاوسي والتنبؤ

```
In  [1]:  from sklearn.svm import SVC
          classifier = SVC(kernel='rbf')
          classifier.fit(X_train, y_train)
In  [2]:  y_pred = classifier.predict(X_test)
```

تقييم الخوارزمية

```
In  [3]:  from sklearn.metrics import classification_report,
          confusion_matrix
          print(confusion_matrix(y_test, y_pred))
          print(classification_report(y_test, y_pred))

Out [7]:  [[15  0  0]
           [ 0 11  0]
           [ 0  0 12]]
                          precision    recall  f1-score   support

             Iris-setosa       1.00      1.00      1.00        15
         Iris-versicolor       1.00      1.00      1.00        11
          Iris-virginica       1.00      1.00      1.00        12

                accuracy                           1.00        38
               macro avg       1.00      1.00      1.00        38
```

```
   weighted avg       1.00      1.00      1.00        38
```

التدريب بالنواة سيگمويد والتنبؤ

```
In  [1]:  from sklearn.svm import SVC
          classifier = SVC(kernel='sigmoid')
          classifier.fit(X_train, y_train)
In  [1]:  y_pred = classifier.predict(X_test)
```

تقييم الخوارزمية

```
In  [3]:  from sklearn.metrics import classification_report,
          confusion_matrix
          print(confusion_matrix(y_test, y_pred))
          print(classification_report(y_test, y_pred))

Out [7]:  [[15  0  0]
           [ 0  7  4]
           [ 0  1 11]]
                         precision    recall  f1-score   support

              Iris-setosa       1.00      1.00      1.00        15
          Iris-versicolor       0.88      0.64      0.74        11
           Iris-virginica       0.73      0.92      0.81        12

                 accuracy                           0.87        38
                macro avg       0.87      0.85      0.85        38
             weighted avg       0.88      0.87      0.87        38
```

مقارنة أداء النوى

إذا قارنا أداء أنواع مختلفة من النوى ، فيمكن ملاحظة أن أداء نواة سيگمويد كان أسوأ من النوى الأخرى. هذا لأن دالة سيگمويد ترجع القيمتين 0 و 1 ، لذا فهي أكثر ملاءمة لمشاكل التصنيف الثنائي. بين النواة الغاوسية والنواة الخطية ، يمكننا أن نرى أن نواة غاوس قد وصلت إلى معدل تنبؤ كامل بنسبة 100٪ ، بينما أخطأت النواة الخطية في تصنيف عينة. لذلك ، كان أداء النواة الغاوسية أفضل. ومع ذلك ، لا توجد قاعدة عامة حول أي نواة تؤدي بشكل أفضل في أي سيناريو. لذلك ، فقط عن طريق اختبار نوى مختلفة ومراقبة النتائج يمكن تحديد نوع النواة في كل مشكلة.

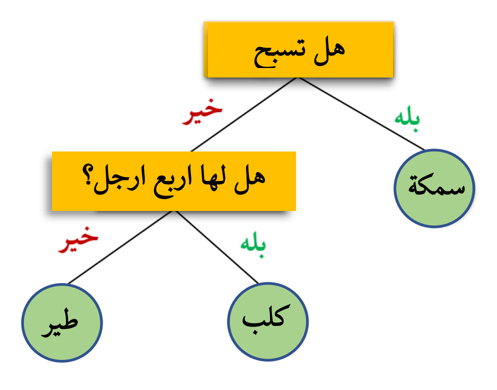
شجرة القرار

واحدة من أكثر خوارزميات التعلم الآلي شيوعًا هي أشجار القرار نظرًا لعملياتها البسيطة جدًا. على عكس آلة المتجه الداعمة ، والتي تتطلب أساسًا رياضيًا قويًا للغاية لفهمها ، فإن أشجار القرار تحاكي حرفياً الطريقة التي نقوم بها نحن البشر على أساس يومي. على سبيل المثال: لنفترض أن لدينا غلاية ونريد أن نحملها ، لكن في نفس الوقت ، لا نريد حرق أيدينا أو إسقاطها لأنها ثقيلة.

في تحليل القرار ، يمكن استخدام أشجار القرار لتمثيل القرارات واتخاذ القرارات بشكل موضوعي وصريح ، وكما يوحي الاسم ، يتم استخدام نموذج يشبه الشجرة للوصول إلى القرار النهائي. على الرغم من أنها أداة شائعة في التنقيب في البيانات لاستخراج استراتيجية لتحقيق هدف معين ، إلا أنها تستخدم على نطاق واسع في التعلم الآلي.

في التعلم الآلي ، تعتبر أشجار القرار نموذجًا غير ابراتوري يمكن استخدامه لكل من التصنيف والانحدار. هذا يعني أن أشجار القرار عبارة عن نماذج مرنة لا تزيد من عدد المعاملات عن طريق إضافة المزيد من الميزات (إذا قمنا ببنائها بشكل صحيح) ويمكن أن تكون تنبوًا فئويًا (مثل ما إذا كان النبات من نوع معين أم لا) أو حتى ينتج التنبؤ العددي (مثل سعر المنزل).

تم اقتراح أشجار القرار لأول مرة من قبل ليو بريمن ، خبير الإحصاء في جامعة كاليفورنيا ، بيركلي. كانت فكرته هي عرض البيانات كشجرة تمثل فيها كل عقدة داخلية تجربة على خاصية (أساسًا عبارة عن شرط) ، ويمثل كل فرع نتيجة الاختبار ، ولكل عقدة ورقية (عقدة طرفية) تسمية فئة. باختصار ، كل سؤال يتم طرحه في عملية اتخاذ القرار هو تجربة على ميزة ، ويؤدي كل اختبار إما إلى استنتاج أو اختبار إضافي يخضع للإجابة الحالية. بشكل أساسي ، تتعلم أشجار القرار مجموعة من القواعد الصريحة حول قيم السمات التي تؤدي إلى قرار يتنبأ بقيمة الهدف. الشكل التالي هو مثال لشجرة قرار بسيطة تستخدم لتصنيف حيوان كطائر أو كلب أو سمكة بناءً على خصائصها (السباحة أو من ذوات الاربع).:



تطرح خوارزمية شجرة القرار لتحليل الفئات الثلاث وصياغة الأسئلة اللازمة للتمييز بين الفئات المختلفة في المثال أعلاه هذه الأسئلة:

- هل تسبح؟
- هل لها اربع ارجل؟

عندما تتم الإجابة على هذه الأسئلة بنعم أو لا ، فإننا نعرف بالتأكيد الفئة التي ينتمي إليها الحيوان. ومن ثم، فإن أشجار القرار لديها مفهوم بسيط نسبيًا لفهمه. لأنها بديهية للغاية.

عادةً ما تتكون شجرة القرار من عقدة جذر والعديد من العقد الداخلية والعديد من العقد الطرفية. تتوافق العقد الطرفية مع نتائج القرار ، وتتوافق كل عقدة أخرى مع اختبار الخاصية. يتم تقسيم العينات في كل عقدة إلى عقد فرعية وفقًا لنتائج التفرع المتفرّع. كل مسار من العقدة الجذرية إلى العقدة الطرفية هو تسلسل قرار. تحاول أشجار القرار تفريع مجموعة البيانات بطريقة تجعل البيانات في كل مجموعة متشابهة قدر الإمكان ، بينما تختلف البيانات في مجموعة واحدة قدر الإمكان عن البيانات في المجموعات الأخرى. الهدف هو إنتاج شجرة يمكنها تعميم العينات غير المرئية.

يتبع بناء أشجار القرار استراتيجية **فرق تسد**[1]، ومن ثم يتم إنتاج شجرة القرار للخلف (لأعلى ولأسفل). بعبارات أبسط ، تبدأ خوارزمية شجرة القرار بمجموعة بيانات التدريب في عقدة الجذر وتقسم البيانات بشكل متكرر إلى عقد ذات مستوى أدنى بناءً على معيار التفرع. فقط العقد التي تحتوي على مجموعة من الفئات المختلفة يجب أن تكون متفرعة أكثر. أخيرًا ، توقف خوارزمية الشجرة قرار نمو الشجرة بناءً على **معيار التقسيم**[2]. أبسط معيار تقسيم هو المعيار الذي تنتمي فيه جميع الأمثلة التدريبية على الورقة إلى نفس الفئة. تتمثل إحدى المشكلات في أن بناء شجرة قرار على هذا المستوى قد يؤدي إلى overfitting. مثل هذه الشجرة لا تعمم جيداً على العينات التجريبية غير المرئية. يستخدم آلية **التقليم**[3] لمنع تقليل الدقة بسبب overfitting.

بمجرد إنشاء شجرة القرار ، يتم استخدامها لتصنيف عينات الاختبار باستخدام التمرير لأسفل من الجذر إلى الورقة الفردية. يتم استخدام حالة الفرع في كل عقدة داخلية لتحديد الفرع الصحيح لشجرة القرار لمزيد من التمرير ، ويتم الإبلاغ عن تسمية العقدة الطرفية التي تم الوصول إليها لعينة الاختبار.

الميزة الرئيسية لشجرة القرار هى أنها لا تحتاج إلى تحجيم البيانات وتوحيدها.

---

[1] Divide-and-conquer

[2] split criterion

[3] pruning

معايير اختيار الميزة (التفرع الأمثل) وعملية تدريب شجرة القرار

تدريب شجرة القرار هو عملية من أعلى إلى أسفل يتم فيها تقسيم مجموعة بيانات التدريب بشكل متكرر إلى مجموعات فرعية أصغر. يتم تحديد هذه المجموعات الفرعية عن طريق تحديد قيود الميزة في كل مرحلة والتي هي أفضل قدرة على تفريع مجموعة العينة بناءً على معيار محدد يسمى معيار التفرع. في صميم خوارزمية تعلم شجرة القرار هو اختيار خاصية الفرع المثلى. بشكل عام ، مع تقدم عملية التفرع ، نتمنى أن تنتمي المزيد من الحالات في كل عقدة إلى فئة واحدة. الغرض من معيار التفرع هو تعظيم الفصل بين الفئات المختلفة بين العقد الفرعية.

الفكرة الأساسية لخوارزمية شجرة القرار هي تحديد الميزات التي تحتوي على معظم المعلومات حول الخاصية الهدف ثم تقسيم مجموعة البيانات على طول قيم هذه الخصائص. الميزة التي تفصل عدم اليقين بشكل أفضل عن المعلومات حول الميزة الهدف هي الميزة الأكثر إفادة. يستمر البحث عن الميزة الأكثر إفادة حتى تصل إلى العقد الورقية النقية. تتضمن عملية بناء نموذج شجرة القرار طرح سؤال واحد في كل حالة ثم المتابعة والتفرع. الآن عندما يكون هناك العديد من السمات التي تحدد القيمة المستهدفة لعينة معينة ، تظهر الأسئلة التالية:

- **ما هي السمة التي يجب تحديدها في عقدة الجذر للبدء؟**
- **كيف ننتقل إلى تحديد الميزات في كل فرع لاحق في العقدة؟**
- **ما الميزة التي ستعمل كعقدة داخلية أو عقدة طرفية ؟**

لاتخاذ قرار بشأن هذه الحالات وكيفية تفريع الشجرة ، نستخدم معايير التفرع. فيما يلي وصف لمعايير التفرع الأكثر شمولاً المستخدمة في تعليم أشجار القرار.

كسب المعلومات

يعتمد هذا المعيار على مفهوم إنتروبيا المعلومات، الذي يقيس الفوضى أو عدم اليقينفي النظام. كسب المعلومات هو قياس تغيرات الانتروبيا بعد تقسيم مجموعة بيانات بناءً على سمة وحساب مقدار المعلومات التي توفرها إحدى السمات حول فئة ما. اعتمادًا على مقدار كسب المعلومات، فإنه يقسم العقدة ويشكل شجرة القرار. تحاول خوارزمية شجرة القرار دائمًا تعظيم مقدار الحصول على المعلومات وتقسيم العقدة أو السمة التي تحتوي على أعلى قدر من المعلومات أولاً.

لنفترض أن p_k يمثل نسبة الفئة k في مجموعة البيانات D و $k = 1,2, \dots, |y|$. وفقًا لذلك، يتم تعريف الانتروبيا على النحو التالي:

$$Ent(D) - \sum_{k=1}^{|y|} p_k log_2 \, p_k \qquad (7_6)$$

كلما انخفض $Ent(D)$ ، زاد نقاء D.

افترض أن الخاصية المنفصلة a لها قيم V المحتملة $\{a^1, a^2, \dots, a^V\}$.. وفقًا لذلك، ينتج عن تقسيم مجموعة D استنادًا إلى الخاصية a عقدة الابن V ، حيث يحتوي V على عقدة الابن D^V التي تحتوي على جميع الحالات في D التي تأخذ القيمة a^V للخاصية a. بعد ذلك، يمكن حساب الانتروبيا D^V باستخدام المعادلة (6ـ7). نظرًا لوجود عينات مختلفة في العقد الفرعية، فإن الوزن $\frac{|D^v|}{|D|}$ وهو مخصص لإظهار أهمية كل عقدة، أي كلما زاد عدد العينات، زاد تأثير العقدة الفرعية. ومن ثم، يتم حساب كسب المعلومات من تفرع مجموعة البيانات D بالخاصية a باستخدام المعادلة التالية:

$$Gain(D, a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

بشكل عام، كلما زاد كسب المعلومات مع التفرع D والخاصية a ، زاد النقاء الذي يمكننا توقعه..

إنه مجتمع نقي ينتمي فيه جميع الأعضاء إلى نفس المجموعة.

المعامل الجيني

المعامل الجيني هو مقياس للشوائب أو النقاء المستخدم لإنشاء حدود القرار. يجب تفضيل الخاصية ذات معامل جيني المنخفض على مؤشر جيني العالي. باستخدام نفس الرمز (7ـ6) ، يتم تعريف قيمة Gini لمجموعة D على النحو التالي:

$$Gini(D) = \sum_{k=1}^{|y|} \sum_{k' \neq k} p_k \, p_{k'}$$

$$= 1 - \sum_{k=1}^{|y|} p_k^2$$

بشكل حدسي ، تمثل $Gini(D)$ احتمال وجود عينتين اخترناهما عشوائيًا من مجموعة بيانات D التي تنتمي إلى فئات مختلفة. كلما انخفض عامل جيني $Gini(D)$ ، زادت درجة نقاء مجموعة بيانات D.

باستخدام رمز مشابه لكسب المعلومات ، يتم تعريف عامل جيني للسمة a على النحو التالي:

$$Gini_index(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} gini(D^v)$$

وفقًا لمجموعة ميزات المرشح A ، نختار الميزة ذات عامل جيني الأدنى كميزة الفرع.

معيار التوقف والتقليم

يرتبط معيار توقف نمو شجرة القرار ارتباطًا وثيقًا باستراتيجية التقليم. عندما تنمو شجرة القرار حتى النهاية ، أي ما دامت كل عقدة في الورقة تحتوي فقط على عينات تنتمي إلى فئة معينة ، تظهر شجرة القرار الناتجة دقة بنسبة 100٪ في العينات التي تنتمي إلى البيانات التدريبية. ومع ذلك ، في هذه الحالة ، غالبًا ما يتم تعميمها بشكل سيئ على عينات الاختبار غير المرئية. هذا لأن شجرة القرار متوافقة الآن حتى مع الميزات العشوائية في الأمثلة التدريبية. تحدث معظم هذه الضوضاء بسبب العقد ذات المستوى الأدنى التي تحتوي على نقاط بيانات أقل. ومن ثم ، تُفضل النماذج الأبسط (أشجار القرار الضحلة) على النماذج الأكثر تعقيدًا (أشجار القرار العميق) إذا ارتكبت نفس الخطأ في بيانات التدريب.

لتقليل مستوى overfitting ، تتمثل إحدى الإستراتيجيات في منع الشجرة من النمو عاجلاً. لسوء الحظ ، لا توجد طريقة لمعرفة النقطة الدقيقة التي تتوقف عندها الشجرة عن النمو. لذلك، تتمثل الإستراتيجية الطبيعية في تقليم الأجزاء المتضخمة من شجرة القرار وتحويل العقد الداخلية إلى عقد ورقية. تشمل استراتيجيات التقليم العامة **التقليم المسبق**[1] **وما بعد التقليم**[2].

يقوم التقليم المسبق بتقييم تحسين القدرة على تعميم كل فرع ، وإذا كان التحسن صغيرًا ، فإنه يلغي الفرع ، أي يتم تحديد العقدة كعقدة ورقية. في المقابل ، يعيد التقليم اللاحق فحص العقد غير الورقية لشجرة قرار مطورة بالكامل ، وإذا أدى البديل إلى تحسين قدرة التعميم ، يتم استبدال عقدة واحدة بالعقدة الورقية.

يتميز التقليم المسبق بأنه أسرع وأكثر كفاءة ، لأنه يتجنب إنشاء أشجار فرعية شديدة التعقيد تتوافق مع بيانات التدريب. في مرحلة ما بعد التقليم ، تقوم بزراعة الشجرة بالكامل باستخدام خوارزمية شجرة القرار الخاصة بك ثم تقليم الأشجار الفرعية لأعلى ولأسفل. تبدأ بعقدة القرار السفلية وتقرر معايير مثل كسب المعلومات ما إذا كنت تريد الحفاظ على عقدة القرار هذه أو استبدالها بعقدة طرفية.

التقليم المسبق:

- يتم استخدام هذه التقنية قبل بناء شجرة القرار.
- يمكن إجراء التقليم المسبق عن طريق ضبط المعاملات الفائقة.

مابعد التقليم:

- تستخدم هذه التقنية بعد صنع شجرة القرار.
- تُستخدم هذه التقنية عندما يكون لشجرة القرار عمق كبير جدًا وتُظهر نموذجًا كبيرًا.

[1] pre-pruning

[2] post-pruning

- تُعرف هذه التقنية أيضًا باسم **التقليم الخلفي**[1].

مميزات شجرة القرار:

- مقاومة للخطأ ، وإذا كانت بيانات التدريب تحتوي على أخطاء ، فإن خوارزميات شجرة القرار ستكون أكثر ملاءمة لمعالجة مثل هذه المشكلات.
- تنتج قواعد مفهومة وبديهية للغاية.
- يمكنها إدارة المتغيرات المستمرة والمتقطعة.
- طريقة عملها بسيطة للغاية ويمكن شرحها بسهولة لأي شخص.
- ليس لديها أي افتراضات حول خطية البيانات وبالتالي يمكن استخدامها عندما تكون المعاملات غير الخطية ذات صلة.
- توفر وقت إعداد البيانات ، لأنها ليست حساسة للقيم المفقودة والقيم البعيدة.
- ليست هناك حاجة لتوحيد وتحجيم البيانات.
- تعتبر مفهوم شجرة القرار أكثر دراية للمبرمجين وأسهل في الفهم من الخوارزميات المماثلة الأخرى.

عيوب شجرة القرار:

- عرضة للأخطاء في مشاكل التصنيف مع الفصول الكبيرة وعدد صغير نسبيًا من الأمثلة التعليمية.
- بالنسبة لشجرة القرار ، قد يكون الحساب أحيانًا أكثر تعقيدًا من الخوارزميات الأخرى.
- غالبًا ما يستغرق تدريب النموذج مزيدًا من الوقت.
- يمكن أن تؤدي إضافة نقطة بيانات جديدة إلى إعادة إنتاج الشجرة ، ويجب إعادة حساب جميع العقد وإنشائها.
- غالبًا ما تكون شجرة القرار المنفردة متعلمًا ضعيفًا ، لذلك من أجل التنبؤ الأفضل ، نحتاج إلى مجموعة من أشجار القرار لإنشاء غابة عشوائية.
- عرضة للضبط الزائد. من أجل ملاءمة البيانات (حتى بيانات الضوضاء) ، تستمر في إنشاء عقد جديدة ، وفي النهاية تصبح الشجرة معقدة للغاية بحيث لا يمكن تفسيرها. بهذه الطريقة يفقد قابليته للتعميم.

---

[1] backward pruning

- بسبب الضبط الزائد ، هناك احتمال كبير للتباين (من أجل تحقيق صفر تحيز ، مما يؤدي إلى تباين كبير) في المخرجات ، مما يؤدي إلى العديد من الأخطاء في التقدير النهائي ويظهر دقة منخفضة في النتائج.

شجرة القرار في بايثون

مجموعة البيانات

في هذا المثال ، نستخدم أيضًا مجموعة بيانات iris. في الأمثلة السابقة ، وصلنا إلى مجموعة البيانات هذه من خلال موقعها على الإنترنت. مجموعة البيانات هذه متاحة أيضًا في مكتبة Scikit-Learn. في هذا المثال ، نقوم باستيراده من خلال المكتبة.

استيراد المكتبات

```
In [1]:  import numpy as np
         import matplotlib.pyplot as plt
         import pandas as pd
```

استيراد مجموعة البيانات

```
In [2]:  from sklearn import datasets
         iris = datasets.load_iris()
```

تحضير البيانات

تتمثل الخطوة التالية في تقسيم مجموعة البيانات إلى سماتها وعلاماتها. استخدم الكود التالي للقيام بذلك:

```
In [4]:  X = iris.data
         y = iris.target
```

تقسيم مجموعة البيانات

```
In [5]:  from sklearn.model_selection import train_test_split

         X_train, X_test, y_train, y_test = train_test_split(X, y,
         test_size =0.25, random_state=42)
```

التدريب والتنبؤ

تتضمن مكتبة شجرة Scikit-Learn طرقًا لمختلف خوارزميات شجرة القرار. نظرًا لأننا سنقوم بمهمة تصنيف في هذا المثال ، فإننا نستخدم كلاس DecisionTreeClassifier لهذا المثال.

```
In [1]: from sklearn.tree import DecisionTreeClassifier
        classifier = DecisionTreeClassifier()
        classifier.fit(X_train, y_train)
```

الآن وقد تم تدريب المصنف الخاص بنا ، فلنقم ببعض التنبؤات حول بيانات الاختبار. للقيام بذلك ، قم بتشغيل التعليمات البرمجية التالية:

```
In [1]: y_pred = classifier.predict(X_test)
```

تقييم الخوارزمية

في هذه الخطوة ، قمنا بتدريب الخوارزمية الخاصة بنا وقمنا ببعض التنبؤات. الآن نريد أن نرى مدى دقة الخوارزمية لدينا.

```
In [3]: from sklearn.metrics import classification_report,
        confusion_matrix
        print(confusion_matrix(y_test, y_pred))
        print(classification_report(y_test, y_pred))

Out [3]: [[15  0  0]
          [ 0 11  0]
          [ 0  0 12]]
                       precision    recall  f1-score   support

           Iris-setosa       1.00      1.00      1.00        15
       Iris-versicolor       1.00      1.00      1.00        11
        Iris-virginica       1.00      1.00      1.00        12

              accuracy                           1.00        38
             macro avg       1.00      1.00      1.00        38
          weighted avg       1.00      1.00      1.00        38
```

تظهر النتائج أن نموذج شجرة القرار لدينا قادر على تصنيف جميع السجلات الـ 38 في مجموعة الاختبار بدقة 100٪.

طباعة التمثيل كنص

يمكن أن يكون ناتج شجرة القرار كتمثيل نصي مفيداً عند العمل على تطبيقات بدون واجهة مستخدم او وقت ، أو عندما نريد تخزين معلومات النموذج في ملف نصي. للقيام بذلك ، قم بتنفيذ التعليمات البرمجية التالية:

```
In [3]: from sklearn import tree
        text_representation = tree.export_text(classifier)
```

```
print(text_representation)
```

Out [3]:
```
|--- feature_3 <= 0.80
|   |--- class: 0
|--- feature_3 >  0.80
|   |--- feature_2 <= 4.75
|   |   |--- feature_3 <= 1.65
|   |   |   |--- class: 1
|   |   |--- feature_3 >  1.65
|   |   |   |--- class: 2
|   |--- feature_2 >  4.75
|   |   |--- feature_3 <= 1.75
|   |   |   |--- feature_2 <= 4.95
|   |   |   |   |--- class: 1
|   |   |   |--- feature_2 >  4.95
|   |   |   |   |--- feature_3 <= 1.55
|   |   |   |   |   |--- class: 2
|   |   |   |   |--- feature_3 >  1.55
|   |   |   |   |   |--- feature_2 <= 5.45
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_2 >  5.45
|   |   |   |   |   |   |--- class: 2
|   |   |--- feature_3 >  1.75
|   |   |   |--- feature_2 <= 4.85
|   |   |   |   |--- feature_1 <= 3.10
|   |   |   |   |   |--- class: 2
|   |   |   |   |--- feature_1 >  3.10
|   |   |   |   |   |--- class: 1
|   |   |   |--- feature_2 >  4.85
|   |   |   |   |--- class: 2
```

إذا كنت تريد حفظه في ملف ، فقم بتنفيذ الكود التالي:

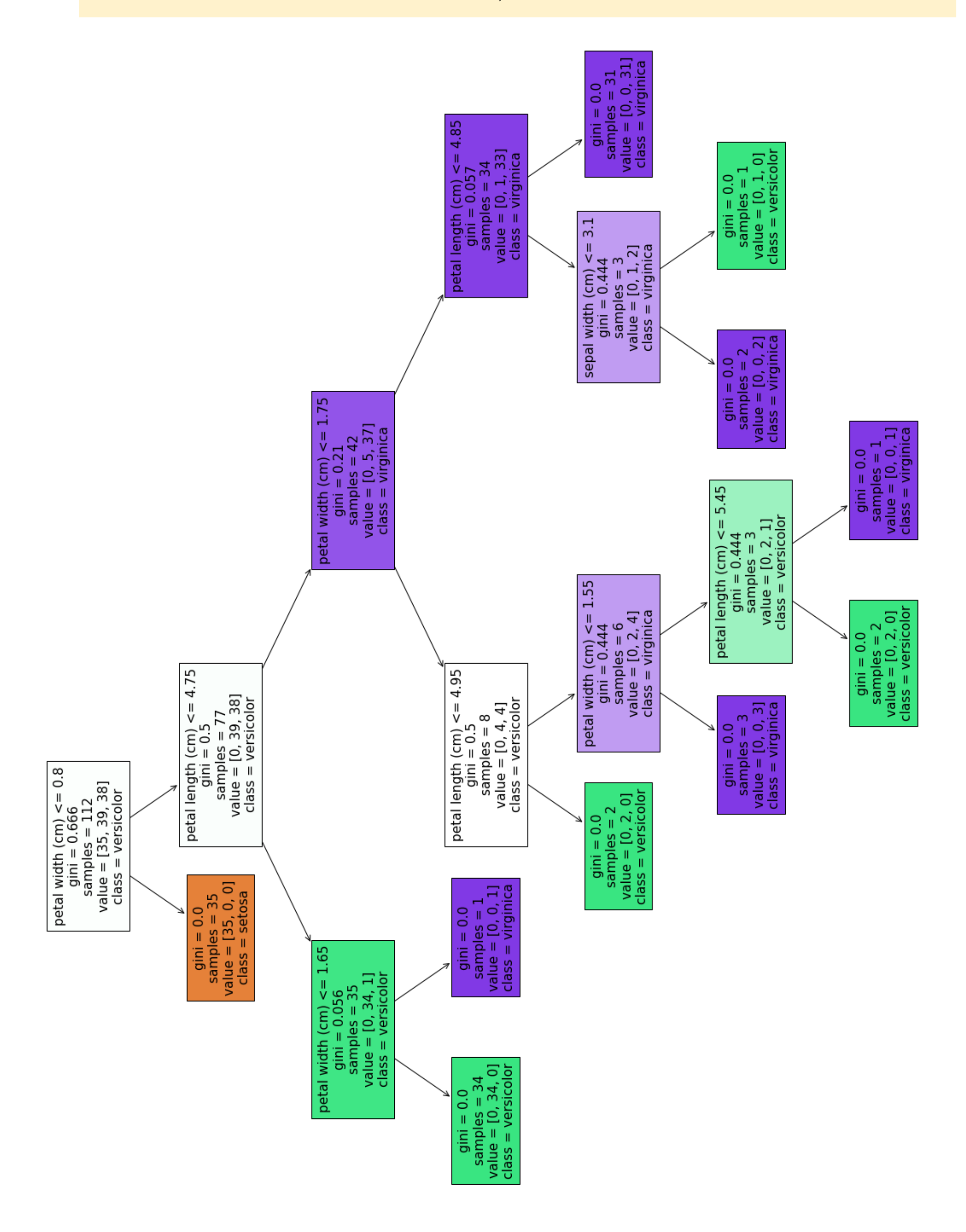
In [3]:
```
with open("decistion_tree.log", "w") as fout:
    fout.write(text_representation)
```

التمثيل المرئي للنموذج

هناك العديد من المكتبات للتمثيل المرئي لشجرة القرار. ومع ذلك ، في هذا المثال ، نستخدم طريقة plot_tree في مكتبة Scikit-Learn. تسمح لنا هذه الطريقة بإنتاج شكل الشجرة بسهولة. للقيام بذلك ، قم بتنفيذ التعليمات البرمجية التالية:

In [3]:
```
fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(classifier,
```

```
feature_names=iris.feature_names,
class_names=iris.target_names,
filled=True)
```



نايف بايز

نايف بايز Naive Bayes هي تقنية تصنيف تستند إلى نظرية بايز ، بافتراض أن جميع الخصائص التي تتنبأ بالقيمة المستهدفة مستقلة عن بعضها البعض. تحسب هذه التقنية احتمالية كل فئة ثم تحدد الفئة الأكثر احتمالية. لكن لماذا تسمى الخوارزمية "ساذجة"؟ هذا لأن المصنف يفترض أن خصائص الإدخال التي تدخل النموذج مستقلة عن بعضها البعض. وبالتالي ، لا يؤثر تغيير ميزة إدخال واحدة على أي ميزة أخرى. بسيط جدًا ، بمعنى أن هذا الافتراض قد يكون أو لا يكون صحيحًا.

الأساس لنظرية نايف بايز أو قاعدة بايزي هو $p(H|S) = \frac{p(S|H)p(S)}{p(H)}$، والتي تقدر **الاحتمال البعدي**[1] $p(H|S)$ لـ a الفرضية أو النموذج. هنا $p(S|H)$ هو **احتمال الامكان**[2] من صحة عينات البيانات ؛ بالنظر إلى أن H صحيح وأن $p(H)$ هو الاحتمال السابق[3] (الأساسي) لفرضية H ، والذي يتضمن بطريقة ما أي **معرفة مسبقة**[4] بـ H. إذا لم تكن المعرفة السابقة متوفرة ، فيمكننا استخدام توزيعات موحدة كما كان من قبل. بالإضافة إلى ذلك ، يمكن اعتبار $p(S)$ بمثابة الاحتمال السابق لعينة S. اعتمادًا على التركيز والصياغة ، يمكننا البحث في الحد الاقصى للفرضية (MAP).

$$maximize\ p(H|S)\ \propto\ maximize\ P(S|H)P(H)$$

أو يمكننا البحث عن **الاحتمال الارجح**[5] **(ML)**

$$maximize\ p(S|H)$$

إذا كانت كل من $p(H)$ و $p(S)$ ثابتة ، فإنها تعادل MAP السابقة.

بالنسبة لمشكلة التصنيف التي تركز على الخاصية $F = x$ مع K قيم الميزات المختلفة $[x_1, x_2, \dots, x_k]$ تشكل دالة النموذج $y = f(x)$ مجموعة من القيم المحددة المتقطعة $y_i \in \omega$. الغرض من المصنف البايزي هو تقدير الاحتمال y ببيانات x_i لإسناد احتمال الفئة

$$\max p(y_i|x_i) = p(y_i|x_1, x_2, \dots, x_k)\ , y_i \in \omega$$

وهو ما يعادل:

[1] posterior probability

[2] likelihood probability

[3] prior probability

[4] background knowledge

[5] maximum likelihood (ML)

$$\max \frac{p(x_1, x_2, \ldots.., x_k | y_i) p(y_i)}{p(x_1, x_2, \ldots.., x_k)} \propto \max p(x_1, x_2, \ldots.., x_k | y_i) p(y_i)$$

ومع ذلك ، فإن حساب الاحتمال $p(x_1, x_2, \ldots.., x_k | y_i)$ هو أمر صعب وفي معظم الحالات يستحيل حسابه. الافتراض البسيط هو أن جميع قيم بيانات العينة مستقلة بشكل مشروط عن بعضها البعض ، وبالتالي فإن الاحتمال المشترك هو نتاج الاحتمال الفردي. حتى نتمكن من استخدامه:

$$p(x_1, x_2, \ldots.., x_k | y_i) p(y_i) = \prod_{i=1}^{k} p(x_i | y_i)$$

وهو ما يعادل:

$$\max p(y_i) \prod_{i=1}^{k} p(x_i | y_i)$$

المصنف الاحتمالي الذي يستخدم المعادلة أعلاه لتعيين الاحتمالات يصبح مصنفًا نايف بايز.

لتعليم مصنف نايف بايز ، نحسب الاحتمال السابق $p(y_i)$ من مجموعة التدريب D ، ثم نحسب الاحتمال الشرطي $p(x_i | y_i)$ لكل ميزة. إذاكان D_{y_i} يمثل مجموعة فرعية من D تحتوي على جميع عينات الفئة y_i. بافتراض أن العينات **مستقلة عن نفس التوزيع**[1] (i.i.d) ، يمكن تقدير الاحتمال السابق بسهولة.

$$p(y_i) = \frac{|D_{y_i}|}{|D|}$$

بالنسبة للخصائص المتقطعة ، إذا كانت $D_{y_{i,}x_i}$ تمثل مجموعة فرعية من D_{y_i} تحتوي على جميع العينات التي تتلقى قيمة x_i في الخاصية ith ، فيمكن تحقيق الاحتمال الشرطي $p(x_i | y_i)$ بواسطة

$$p(x_i | y_i) = \frac{|D_{y_{i,}x_i}|}{|D_{y_i}|}$$

في بعض الحالات ، خاصة في مجموعة البيانات المستمرة ، يمكن افتراض أن العينات مأخوذة من التوزيع الغاوسي. ومن ثم ، فإننا نفترض أن $p(x_i | y_i) \sim N(\mu_{y_i}, \sigma_{y_i}^2)$، حيث μ μ_{y_i} و $\sigma_{y_i}^2$ هي متوسط وتباين خاصية i-th لـ عينات الفئة ، على التوالي. لدينا وفقا لذلك:

---

[1] Independent and identically distributed

$$p(x_i|y_i) = \frac{1}{\sqrt{2\pi}\sigma_{y_i}} \exp\left(-\frac{(x_i - \mu_{y_i})^2}{2\sigma_{y_i}^2}\right).$$

الآن ، لنقم بتدريب تصنيف نايف بايز باستخدام مجموعة بيانات البطيخ في الجدول (6–1) وصنف البطيخ T1 التالي:

ID	color	root	sound	texture	umbilicus	surface	density	sugar	ripe
T1	green	curly	muffled	clear	hollow	hard	0.697	0.460	?

جدول 6ــ1 مجموعه بيانات البطيخ (*Zhou, 2021*)

ID	**color**	**root**	**sound**	**texture**	**umbilicus**	**surface**	**density**	**sugar**	**ripe**
1	green	curly	muffled	clear	hollow	hard	0.697	0,460	true
2	dark	curly	dull	clear	hollow	hard	0.774	0,376	true
3	dark	curly	muffled	clear	hollow	hard	0.634	0,264	true
4	green	curly	dull	clear	hollow	hard	0,608	0,318	true
5	light	curly	muffled	clear	hollow	hard	0,556	0,215	true
6	green	slightly curly	muffled	clear	slightly hollow	soft	0,403	0,237	true
7	dark	slightly curly	muffled	slightly blurry	slightly hollow	soft	0,481	0,149	true
8	dark	slightly curly	muffled	clear	slightly hollow	hard	0,437	0,211	true
9	dark	slightly curly	dull	slightly blurry	slightly hollow	hard	0,666	0,091	false
10	green	straight	dull	clear	flat	soft	0,243	0,267	false
11	light	straight	dull	blurry	flat	hard	0,245	0,057	false
12	light	curly	muffled	blurry	flat	soft	0,343	0,099	false
13	green	slightly curly	muffled	slightly blurry	hollow	hard	0,639	0,161	false
14	light	slightly curly	dull	slightly blurry	hollow	hard	0,657	0,198	false
15	dark	slightly curly	muffled	clear	slightly hollow	soft	0,360	0,370	false
16	light	curly	muffled	blurry	flat	hard	0,593	0,042	false
17	green	curly	dull	slightly blurry	slightly hollow	hard	0,719	0,103	false

نحن أولا نقدر (نخمن) الاحتمال:

$$P(ripe = true) = \frac{8}{17} \approx 0.471$$

$$P(ripe = false) = \frac{9}{17} \approx 0.529$$

بعد ذلك ، نقدر الاحتمال الشرطي لكل خاصية $p(x_i|y_i)$:

$$P_{green|true} = P(color = green|ripe = true) = \frac{3}{8} = 0.375$$

$$P_{green|false} = P(color = green|ripe = false) = \frac{3}{9} \approx 0.333$$

$$P_{curly|true} = P(root = curly|ripe = true) = \frac{5}{8} = 0.625$$

$$P_{curly|false} = P(root = curly|ripe = false) = \frac{3}{9} \approx 0.333$$

$$P_{muffled|true} = P(sound = muffled|ripe = true) = \frac{5}{8} = 0.625$$

$$P_{muffled|false} = P(sound = muffled|ripe = false) = \frac{4}{9} \approx 0.444$$

$$P_{clear|true} = P(texture = clear|ripe = true) = \frac{7}{8} = 0.875$$

$$P_{clear|false} = P(texture = clear|ripe = false) = \frac{2}{9} \approx 0.222$$

$$P_{hollow|true} = P(umbilicus = hollow|ripe = true) = \frac{5}{8} = 0.625$$

$$P_{hollow|false} = P(umbilicus = hollow|ripe = false) = \frac{2}{9} \approx 0.222$$

$$P_{hard|true} = P(surface = hard|ripe = true) = \frac{6}{8} = 0.750$$

$$P_{hard|false} = P(surface = hard|ripe = false) = \frac{6}{9} \approx 0.667$$

$$P_{density:0.697|true} = P(density = 0.697|ripe = true)$$

$$= \frac{1}{\sqrt{2\pi} * 0.129} \exp\left(-\frac{(0.697 - 0.574)^2}{2 * 0.129^2}\right) \approx 1.959$$

$$P_{density:0.697|false} = P(density = 0.697|ripe = true)$$

$$= \frac{1}{\sqrt{2\pi} * 0.195} \exp\left(-\frac{(0.697 - 0.496)^2}{2 * 0.195^2}\right) \approx 1.203$$

$$P_{sugar:0.460|true} = P(sugar = 0.460|ripe = true)$$

$$= \frac{1}{\sqrt{2\pi} * 0.101} \exp\left(-\frac{(0.460 - 0.279)^2}{2 * 0.101^2}\right) \approx 0.788$$

$$P_{sugar:0.460|false} = P(sugar = 0.460|ripe = true)$$

$$= \frac{1}{\sqrt{2\pi} * 0.108} \exp\left(-\frac{(0.460 - 0.154)^2}{2 * 0.108^2}\right) \approx 0.066$$

ومن ثم لدينا:

$$P(ripe = true) \times P_{green|true} \times P_{curly|true} \times P_{muffled|true} \times P_{clear|true} \times P_{hollow|true} \times P_{hard|true} \times P_{density:0.697|true} \times P_{sugar:0.460|true} \approx 0.052$$

$$P(ripe = false) \times P_{green|false} \times P_{curly|false} \times P_{muffled|false} \times P_{clear|false} \times P_{hollow|false} \times P_{hard|false} \times P_{density:0.697|false} \times P_{sugar:0.460|false} \approx 6.80 \times 10^{-5}$$

نظرًا لأن $0.052 > 6.80 \times 10^{-5}$، تحدد مصنف بايزي فئة عينة الاختبار T1 على أنها true.

مميزات نايف بايز:

- إنه سهل التنفيذ ، لأنه يتم حساب الاحتمال فقط.
- لا يتطلب الكثير من بيانات التدريب.
- يمكن أن يعمل بشكل أفضل من النماذج الأخرى إذا استمر افتراض عدم الخطية الشرطي.
- يعمل بشكل أسرع للتنبؤ بفئات البيانات التجريبية.

عيوب نايف بايز:

- الافتراض الشرطي ليس صحيحًا دائمًا.
- في مجموعة البيانات التجريبية ، إذا كانت خاصية السلسلة تحتوي على فئة لم يتم ملاحظتها في مجموعة التدريب ، فإن النموذج يعينها احتمالًا بقيمة 0 (صفر) ولن يكون قادرًا على التنبؤ. غالبًا ما يشار إلى هذا على أنه تردد صفري[1].

[1] Zero Frequency

نايف بايز في بايتون

مجموعة البيانات

على سبيل المثال ، نستخدم مجموعة بيانات iris البسيطة.

استيراد المكتبات

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

استيراد مجموعة البيانات

In [2]:
```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

# Assign colum names to the dataset
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']

# Read dataset to pandas dataframe
dataset = pd.read_csv(url, names=names)
```

المعالجة المسبقة

In [4]:
```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 4].values
```

تقسيم مجموعة البيانات

In [5]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =0.25, random_state=42)
```

تدريج البيانات

In [1]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

التدريب والتنبؤ

أولاً قمنا بتقسيم البيانات إلى مجموعات تدريبية وتجريبية ثم قمنا بتدريج الخصائص على البيانات. حان الوقت الآن لتدريب نايف بايز على بيانات التدريب. في هذه الخطوة من كلاس GaussianNB نستخدم مكتبة sklearn.naive_bayes. استخدمنا هنا نموذجًا غاوسيًا ، وهناك العديد من النماذج الأخرى مثل برنولي ومتعددة الحدود وما إلى ذلك.

```
In [1]: from sklearn.naive_bayes import GaussianNB
        classifier = GaussianNB()
        classifier.fit(X_train, y_train)
```

الخطوة الأخيرة هي التنبؤ بالنموذج المبني على بياناتنا التجريبية.

```
In [1]: y_pred = classifier.predict(X_test)
```

تقييم الخوارزمية

```
In [3]: from sklearn.metrics import classification_report,
        confusion_matrix
        print(confusion_matrix(y_test, y_pred))
        print(classification_report(y_test, y_pred))

Out [7]: [[15  0  0]
          [ 0 11  0]
          [ 0  0 12]]
                       precision    recall  f1-score   support

          Iris-setosa       1.00      1.00      1.00        15
      Iris-versicolor       1.00      1.00      1.00        11
       Iris-virginica       1.00      1.00      1.00        12

             accuracy                           1.00        38
            macro avg       1.00      1.00      1.00        38
         weighted avg       1.00      1.00      1.00        38
```

تظهر النتائج أن نموذج نايف بايز الكاوسي قادر على تصنيف جميع السجلات الـ 38 في مجموعة الاختبار بدقة 100٪.

الانحدار (التوقع)

بشكل عام ، هناك طريقتان لاستخراج المعرفة ، الأولى من خلال الخبراء الميدانيين والثانية باستخدام التعلم الآلي. الخبراء ليسوا مفيدين جداً للكميات الكبيرة من البيانات ، لذلك نستخدم التعلم الآلي للقيام بذلك. تتمثل إحدى الطرق التي يمكننا من خلالها استخدام التعلم الآلي في

تكرار منطق الخبراء في شكل خوارزميات ، ومع ذلك ، فإن هذا أمر شاق للغاية ويستغرق وقتًا طويلاً ومكلفًا ، بالإضافة إلى أنه قد لا يحقق ما نحتاج إليه. ومن ثم ، فإن حل هذه المشكلة هو الانتقال إلى خوارزميات **التعلم الاستقرائي**[1] التي تولد إستراتيجية القيام بمهمة ما بنفسها ولا تتطلب تعليمات منفصلة في كل مرحلة. التصنيف والانحدار هما خوارزميات في التعلم الآلي تقع ضمن هذه الفئة. على عكس عمليات التصنيف ، حيث يتم إجراء محاولات للتنبؤ بتسميات الفئات بشكل منفصل ، تتنبأ نماذج الانحدار بالقيم العددية. بمعنى آخر ، يعد الانحدار مشكلة تعلم خاضعة للإشراف حيث يوجد مدخلات x ومخرج y ، والدالة هي تعلم التطبيق من المدخلات إلى المخرجات. لنفترض أننا نريد أن يكون لدينا نظام يمكنه التنبؤ بسعر السيارة المستعملة. المدخلات وخصائص السيارة مثل العلامة التجارية والسنة والمسافة المقطوعة وغيرها من المعلومات التي نعتقد أنها تؤثر على قيمة السيارة والإنتاج هو سعر السيارة. أو فكر في التنقل مع روبوت متحرك (سيارة ذاتية القيادة) ؛ الإخراج هو الزاوية التي يجب أن تدور فيها عجلة القيادة في كل مرة للتقدم دون الاصطدام بالعقبات والانحرافات ، ويتم توفير المدخلات بواسطة أجهزة استشعار في السيارة مثل كاميرات الفيديو ونظام تحديد المواقع العالمي (GPS) وما إلى ذلك.

كيف يعمل الانحدار

في الانحدار ، نريد كتابة الناتج الرقمي y ، والذي يسمى المتغير التابع ، كدالة للمدخل x ، والذي يسمى المتغير المستقل. افترض أن الناتج هو مجموع دالة الإدخال $f(x)$ ومقدار الخطأ العشوائي كما هو موضح:

$$y = f(x) + \epsilon$$

هنا الدالة $f(x)$ غير معروفة ونريد تقريبها بالمقدر $g(x;\theta)$ الذي يحتوي على مجموعة من المعاملات θ . افترض أن الخطأ العشوائي يتبع التوزيع الطبيعي بمتوسط 0. إذا كانت $x_1, \dots, x_n$ هي عينة عشوائية لمشاهدات متغير الإدخال x و $y_1, \dots, y_n$ القيم المرصودة مرتبطة بمتغير الإخراج y. بعد ذلك ، باستخدام افتراض أن الخطأ يتبع التوزيع الطبيعي ، يمكننا استخدام طريقة تقدير الاحتمالية القصوى لتقدير قيم المعامل θ . يمكن إظهار أن القيم θ التي تعظم دالة الاحتمال هي القيم التي تقلل مجموع المربعات التالية:

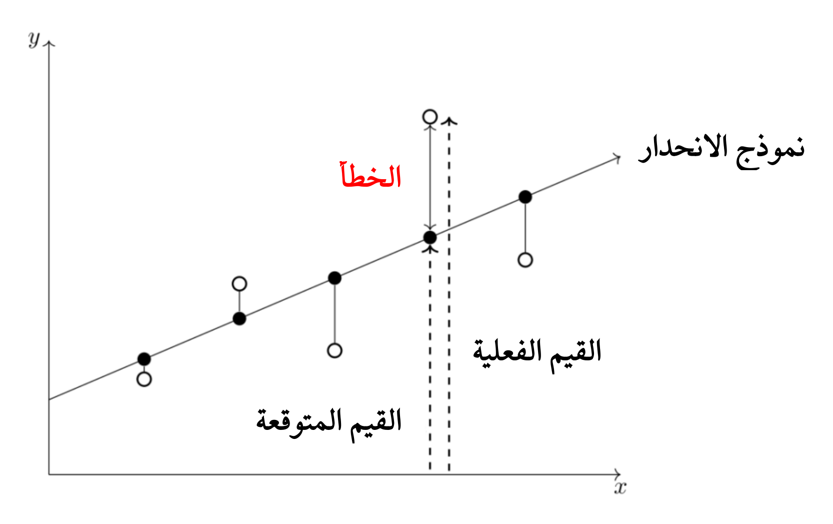
$$E(\theta) = (y_1 - g(x_1,\theta))^2 + \cdots + (y_n - g(x_n,\theta))^2$$

[1] inductive learning

تُعرف طريقة إيجاد قيمة θ كقيمة التي تقلل $E(\theta)$ باسم طريقة **مربعات صغرى عادية**[1] (OLS). في هذا القسم ، نستخدم هذه الطريقة لتقدير المعاملات.

طريقة مربعات صغرى عادية

في طريقة مربعات صغرى عادية، يتم تحديد قيم نقاط التقاطع مع محور y والمنحدر لتقليل مجموع مربعات الأخطاء ؛ أي مجموع مربعات المسافة العمودية بين قيمة y المتوقعة وقيمة y الفعلية (الشكل 6ـ12).



شكل 6ـ12. خطأ في القيم المرصودة

إذا كانت $\hat{y}_i$ هي القيمة المتوقعة y_i، فسيتم إعطاء مجموع مربعات الأخطاء على النحو التالي:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} (y_i - (\alpha + \beta x_i))^2$$

لذلك علينا إيجاد قيمتي α و β و بطريقة تجعل E في حدها الأدنى. يمكن إثبات أنه يمكن الحصول على القيمتين a و b ، وهما القيمتان α و β ، على التوالي ، حيث يكون E هو الحد الأدنى ، عن طريق حل المعادلات التالية:

---

1 Ordinary Least Squares

$$\sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} x_i y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2$$

معادلات ايجاد a و b

للتذكير ، يتم إعطاء متوسط x و y على النحو التالي:

$$\bar{x} = \frac{1}{n}\sum x_i$$

$$\bar{y} = \frac{1}{n}\sum y_i$$

وكذلك التباين x بواسطة المعادلة التالية يُحتسب:

$$Var(x) = \frac{1}{n-1}\sum (x_i - \bar{x})^2$$

يتم تعريف التغاير المشترك بين x و y ، المشار إليهما بواسطة $Cov(x,y)$ على النحو التالي:

$$Cov(x,y) = \frac{1}{n-1}\sum (x_i - \bar{x})(y_i - \bar{y})$$

يمكن إثبات أن قيم b و a يمكن حسابها باستخدام المعادلات التالية:

$$b = \frac{Cov(x,y)}{Var(x)}$$

$$a = \bar{y} - b\bar{x}$$

الانحدار الخطي البسيط

يشير مصطلح "الخطية" في الجبر إلى العلاقة الخطية بين متغيرين أو أكثر ، والتي ، إذا تم رسمها في مساحة ثنائية الأبعاد ، ينتج عنها خط مستقيم. ضع في اعتبارك سيناريو نريد من خلاله تحديد العلاقة الخطية بين عدد ساعات الدراسة للطالب والنسبة المئوية للدرجات التي يحصل عليها الطالب في الاختبار. نريد أن نعرف إلى أي مدى يمكن للطالب أن يحرز درجات عالية بالنظر إلى عدد الساعات التي يعدها للامتحان. إذا قمنا برسم المتغير المستقل (الساعة) على المحور

السيني والمتغير التابع (النسبة المئوية) على المحور الصادي ، فإن الانحدار الخطي يعطينا خطًا مستقيمًا يناسب نقاط البيانات بشكل أفضل.

نتذكر من المدرسة الثانوية أن معادلة الخط المستقيم هي أساسًا على النحو التالي:

$$y = mx + b$$

حيث b هو نقطة التقاطع مع محور y و m هو ميل الخط. لذلك ، تمنحنا خوارزمية الانحدار الخطي القيمة المثلى لنقطة التقاطع مع محور y والمنحدر (في بعدين). تظل المتغيرات y و x ثابتة لأنها خصائص بيانات ولا يمكن تغييرها. ومن ثم ، فإن القيم التي يمكننا التحكم فيها هي نقطة التقاطع مع محور y والميل. يمكن أن يكون هناك عدة خطوط مستقيمة اعتمادًا على قيم نقطة التقاطع مع محور y والميل. ما تفعله خوارزمية الانحدار الخطي بشكل أساسي هو وضع عدة خطوط في نقاط البيانات وإرجاع الخط بأقل خطأ.

لنفترض أن x متغير توقع مستقل وأن y متغير تابع ، وافترض أيضًا أن لدينا مجموعة من القيم المرصودة لـ y و x. يحدد نموذج الانحدار الخطي البسيط العلاقة بين y و x باستخدام خط بواسطة معادلة على النحو التالي:

$$y = \alpha + \beta x$$

من أجل تحديد التقديرات المثلى لـ α و β ، يتم استخدام طريقة تقدير المربعات الأقل شيوعًا الموصوفة سابقًا.

لفهم أفضل ، دعونا نوضح مثالاً. بافتراض أن y متغير مستقل ، نحصل على انحدار خطي للبيانات التالية:

x	1.0	2.0	3.0	4.0	5.0
y	1.00	2.00	1.30	3.75	2.25

لدينا:

$$n = 5$$

$$\bar{x} = \frac{1}{5}(1.0 + 2.0 + 3.0 + 4.0 + 5.0)$$
$$= 3.0$$

$$\bar{y} = \frac{1}{5}(1.00 + 2.00 + 1.30 + 3.75 + 2.25)$$
$$= 2.06$$

$$Cov(x, y) = \frac{1}{4}[(1.0 - 3.0)(1.00 - 2.06) + \cdots + (5.0 - 3.0)(2.25 - 2.06)]$$
$$= 1.0625$$

$$Var(x) = \frac{1}{4}[(1.0 - 3.0)^2 + \cdots + (5.0 - 3.0)^2]$$

$$= 2.5$$

$$b = \frac{1.0625}{2.5}$$

$$= 0.425$$

$$a = 2.06 - 0.425 \times 3.0$$

$$= 0.785$$

لذلك ، يكون نموذج الانحدار الخطي للبيانات على النحو التالي:

$$y = 0.785 + 0.425x$$

يمكنك أيضًا تشغيل الكود التالي في بايثون:

```
In [1]: #data
        X=[[1.0],[2.0],[3.0],[4.0],[5.0]]
        Y=[1.00, 2.00, 1.30, 3.75, 2.25]

        from sklearn.linear_model import LinearRegression
        lr = LinearRegression()
        lr.fit(X, Y)
        #print
        print("b:","%.3f" % round(lr.coef_[0], 3))
        print("a:","%.3f" % round(lr.intercept_, 3))
Out [1]: b: 0.425
         a: 0.785
```

الانحدار متعدد الحدود

لنفترض أن x متغير توقع مستقل و y متغير تابع ، لذلك فإن نموذج الانحدار متعدد الحدود يحدد العلاقة بين y و x بالمعادلة على النحو التالي:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_k x^k$$

يتم استخدام طريقة المربعات الصغرى لتحديد القيم المثلى للمعاملات $\alpha_0, \alpha_1, \ldots, \alpha_k$. قيم المعاملات هي القيم التي تقلل مجموع المربعات:

$$E = \sum_{i=1}^{n} [y_i - (\alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \cdots + \alpha_k x_i^k)]^2$$

يتم الحصول على القيم المثلى للمعاملات من خلال حل نظام المعادلات التالية:

$$\frac{\partial E}{\partial \alpha_i} = 0, i = 0,1, \dots, k.$$

افترض أن قيم المعاملات التي تقلل E هي كما يلي:

$$\alpha_{i=}a_i, i = 0,1, \dots, n.$$

وفقًا لذلك ، يمكن ملاحظة أنه يمكن الحصول على قيم a_i من خلال حل نظام المعادلات الخطية $(k+1)$:

$$\begin{aligned}
\sum y_i &= \alpha_0 n + \alpha_1\left(\sum x_i\right) + \cdots + \alpha_k\left(\sum x_i^k\right)\\
\sum y_i x_i &= \alpha_0\left(\sum x_i\right) + \alpha_1\left(\sum x_i^2\right) + \cdots + \alpha_k\left(\sum x_i^{k+1}\right)\\
\sum y_i x_i^2 &= \alpha_0\left(\sum x_i^2\right) + \alpha_1\left(\sum x_i^3\right) + \cdots + \alpha_k\left(\sum x_i^{k+2}\right)\\
&\vdots\\
\sum y_i x_i^k &= \alpha_0\left(\sum x_i^k\right) + \alpha_1\left(\sum x_i^{k+1}\right) + \cdots + \alpha_k\left(\sum x_i^{2k}\right)
\end{aligned}$$

من خلال حل نظام المعادلات الخطية هذا ، نحصل على القيم المثلى للمعاملات. لفهم أفضل، دعونا نوضح مثالاً. نحن عازمون على إيجاد نموذج انحدار تربيعي للبيانات التالية:

x	3	4	5	6	7
y	2.5	3.2	3.8	6.5	11.5

اسمحوا ان يكون نموذج الانحدار التربيعي على النحو التالي:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2.$$

القيم α_0، α_1 و α_2 التي تقلل من مجموع مربعات الخطأ هي a_0، a_1 و a_2، والتي تحقق المعادلات التالية:

$$\begin{aligned}
\sum y_i &= n a_0 + a_1\left(\sum x_i\right) + a_2\left(\sum x_i^2\right)\\
\sum y_i x_i &= a_0\left(\sum x_i\right) + a_1\left(\sum x_i^2\right) + a_2\left(\sum x_i^3\right)\\
\sum y_i x_i^2 &= a_0\left(\sum x_i^2\right) + a_1\left(\sum x_i^3\right) + a_2\left(\sum x_i^4\right)
\end{aligned}$$

باستخدام البيانات التي قدمت ، لدينا:

$$27.5 = 5a_0 + 25a_1 + 135a_2$$

$$158.8 = 25a_0 + 135a_1 + 775a_2$$

$$966.2 = 135a_0 + 775a_1 + 4659a_2$$

من خلال حل نظام المعادلات ، نحصل على المعادلات:

$$a_0 = 12.4285714$$

$$a_1 = -5.5128571$$

$$a_2 = 0.7642857$$

ومن ثم ، فإن نموذج متعدد الحدود التربيعي المطلوب هو كما يلي:

$$y = 12.4285714 - 5.5128571x + 0.7642857x^2$$

يمكنك أيضًا تشغيل الكود التالي في بايثون:

```
In [1]: #data
        X=[[3],[4],[5],[6],[7]]
        Y=[2.5, 3.2, 3.8, 6.5, 11.5]

        from sklearn.linear_model import LinearRegression
        from sklearn.preprocessing import PolynomialFeatures
        poly_reg = PolynomialFeatures(degree=2)
        X_poly = poly_reg.fit_transform(X)
        lin_reg2 = LinearRegression()
        lin_reg2.fit(X_poly,Y)
        #print
        print(lin_reg2.coef_)
        print(lin_reg2.intercept_)
Out [1]: array([ 0.        , -5.51285714,  0.76428571])
        12.428571428571345
```

الانحدار الخطي المتعدد

لنفترض أن هناك متغيرات N مستقلة $x_1, x_2, \dots, x_N$وأيضًا المتغير y تابع. افترض أيضًا أن هناك قيمًا ملحوظة لهذه المتغيرات:

المتغيرات (السمات)	القيم (العينات)			
	عينة 1	عينة 2	...	عينة n
x_1	x_{11}	x_{12}	...	x_{1n}
x_2	x_{21}	x_{22}	...	x_{2n}
...				
x_N	x_{N1}	x_{N1}	...	x_{Nn}
y	y_1	y_2	...	y_n

يحدد نموذج الانحدار الخطي المتعدد العلاقة بين المتغير المستقل N والمتغير التابع بالمعادلة التالية:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_N x_N$$

كما هو الحال مع الانحدار الخطي البسيط ، نستخدم طريقة المربعات الأقل شيوعًا للحصول على التقديرات المثلى $\beta_0, \beta_1, \ldots, \beta_N$. اذا :

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{N1} \\ 1 & x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & & & & \\ 1 & x_{1n} & x_{2n} & \cdots & x_{Nn} \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, B = \begin{bmatrix} \beta_{.} \\ \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}$$

بعد ذلك ، يمكن إظهار أن معاملات الانحدار يتم تقديمها على النحو التالي:

$$B = (X^T X)^{-1} X^T Y$$

لفهم أفضل ، دعونا نوضح مثالاً. نريد الحصول على نموذج انحدار خطي متعدد للبيانات التالية:

x_1	1	1	2	0
x_2	1	2	2	1
y	3.25	6.5	3.5	5.0

في هذه المشكلة ، يوجد متغيران مستقلان وأربع مجموعات من القيم المتغيرة. ومن ثم ، لدينا $n = 2$ و$N = 4$ للرموز المستخدمة أعلاه. نموذج الانحدار الخطي المتعدد لهذه المشكلة كما يلي:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

الحسابات مبينة أدناه:

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 0 & 1 \end{bmatrix}, Y = \begin{bmatrix} 3.25 \\ 6.5 \\ 3.5 \\ 5.0 \end{bmatrix}, B = \begin{bmatrix} \beta_{.} \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 4 & 4 & 6 \\ 4 & 6 & 7 \\ 6 & 7 & 10 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} \frac{11}{4} & \frac{1}{2} & -2 \\ \frac{1}{2} & 1 & -1 \\ -2 & -1 & 2 \end{bmatrix}$$

$$B = (X^T X)^{-1} X^T Y$$

$$= \begin{bmatrix} 2.0625 \\ -2.3750 \\ 3.2500 \end{bmatrix}$$

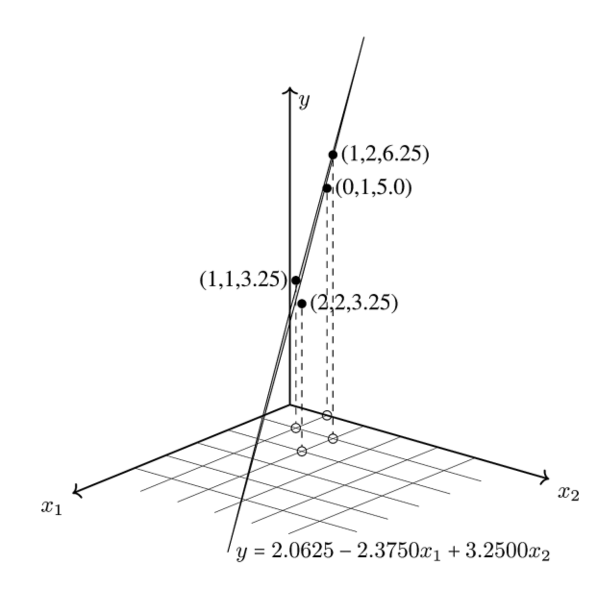
ومن ثم ، فإن النموذج المطلوب هو على النحو التالي:

$$y = 2.0625 - 2.3750x_1 + 3.2500x_2$$

يمكنك أيضًا تشغيل الكود التالي في بايثون:

```
In  [1]:  #data
          X=[[1,1],[1,2],[2,2],[0,1]]
          Y=[3.25, 6.5, 3.5, 5.0]
          from sklearn.linear_model import LinearRegression
          lr = LinearRegression()
          lr.fit(X, Y)
          #print
          print(lr.coef_)
          print(lr.intercept_)
Out [1]:  [-2.375  3.25 ]
          2.0625
```

صفحة الانحدار على سبيل المثال:



خلاصة الفصل السادس

- التصنيف الثنائي ، حيث يتم تخصيص كل عينة لواحدة فقط من الفئتين المحددتين مسبقًا ، هو أبسط أنواع التصنيف.
- التصنيف متعدد العلامات هو شكل معمم للتصنيف ذو علامة واحدة.

- يمكن تصنيف خوارزميات التعلم الآلي إلى فئتين ، بارامتري و لابارامتري.
- أكبر عيب للطرق البارامترية هو أن الافتراضات التي نتخذها قد لا تكون دائمًا صحيحة.
- تُعرف الخوارزميات التي ليس لها افتراضات محددة حول نوع دالة التعيين بالخوارزميات اللابرامترية.
- عندما تبني خوارزمية التعلم الآلي نموذجًا فور تلقي مجموعة بيانات إرشادية ، يطلق عليها اسم المتعلم الطموح.
- عندما لا تبني خوارزمية التعلم الآلي نموذجًا فور تلقي بيانات التدريب ، لكنها تنتظر تقديم بيانات الإدخال للتقييم ، يطلق عليها اسم المتعلم الكسول.
- تعتمد خوارزمية KNN على افتراض أن الأشياء المتشابهة قريبة من بعضها البعض.
- يبحث كي_ اقرب جار في مساحة النمط لعينات التدريب k الأقرب للعينة غير المعروفة.
- يتم استخدام خوارزمية KNN لكل من التصنيف والانحدار.
- تختلف آلات متجهات داعمة عن خوارزميات التصنيف الأخرى من حيث أنها تحدد حدود القرار التي تزيد المسافة من أقرب نقاط البيانات لجميع الفئات.
- غالبًا ما تكون مشكلات العالم الحقيقي غير قابلة للفصل خطيًا ، لذلك لا يمكنك استخدام آلات المتجهات الداعمة الصلبة في هذه المشكلات.
- واحدة من أكثر خوارزميات التعلم الآلي شيوعًا هي أشجار القرار نظرًا لعملياتها البسيطة جدًا.
- أشجار القرار هي نوع من النماذج اللابرامترية.
- تتبع أشجار قرار البناء استراتيجية فرق تسد.
- تعلم شجرة القرار هو عملية من أعلى إلى أسفل.
- تتضمن استراتيجيات التقليم العامة في أشجار القرار التقليم المسبق والتقليم اللاحق.
- يقوم التقليم المسبق بتقييم تحسين القدرة على تعميم كل فرع ، وإذا كان التحسن صغيرًا، يتم إلغاء الفرع.
- يعيد التقليم اللاحق فحص العقد غير الورقية لشجرة قرار مطورة بالكامل ، وإذا أدى الاستبدال إلى تحسين قدرة التعميم ، يتم استبدال عقدة واحدة بعقدة ورقية.
- نايف بايز هي تقنية تصنيف تعتمد على نظرية بايز التي تفترض أن جميع الخصائص التي تتنبأ بالقيمة المستهدفة مستقلة عن بعضها البعض.
- على عكس عمليات التصنيف ، حيث يتم إجراء محاولات للتنبؤ بتسميات الفئات بشكل منفصل ، تتنبأ نماذج الانحدار بالقيم العددية.
- في الانحدار ، نريد كتابة الناتج الرقمي y ، والذي يسمى المتغير التابع ، كدالة للمدخل x ، والذي يسمى المتغير المستقل.

مصادر اضافية لمزيد من القراءة

Lindholm, A., Wahlström, N., Lindsten, F., & Schön, T. B. (2019). Supervised Machine Learning. Lecture notes for the Statistical Machine Learning course.

Kantardzic, M. (2011). Data mining: concepts, models, methods, and algorithms. John Wiley & Sons.

Theodoridis, S. (2015). Machine learning: a Bayesian and optimization perspective. Academic press.

Watt, J., Borhani, R., & Katsaggelos, A. K. (2020). Machine learning refined: Foundations, algorithms, and applications. Cambridge University Press.

Ye, N. (2013). Data mining: theories, algorithms, and examples. CRC press.

7

التعلم العميق

الأهداف:

- ما هو التعلم العميق؟
- ما هو الفرق بين التعلم العميق والتعلم الآلي؟
- متى يستخدم التعلم العميق؟
- عيوب وتحديات التعلم العميق.
- التعرف على الشبكات العصبية العميقة.
- تحسين الشبكة العميقة.

ما هو التعلم العميق؟

التعلم العميق هو مجموعة فرعية من التعلم الآلي الذي يعلم الآلات القيام بالأشياء التي يولد بها البشر بشكل طبيعي: التعلم من خلال الأنماط. على الرغم من أن هذه التقنية غالبًا ما تُعتبر مجموعة من الخوارزميات التي "تحاكي الدماغ" ، فإن الوصف الأكثر ملاءمة هو مجموعة من الخوارزميات التي "تتعلم من خلال الطبقات". بمعنى آخر ، يتضمن التعلم من خلال الطبقات التي تمكن الخوارزمية من إنشاء تسلسل هرمي للمفاهيم المعقدة من المفاهيم الأبسط. لقد جذب هذا المجال الكثير من الاهتمام في السنوات الأخيرة ، وهناك أسباب وجيهة لذلك. لأن التطورات الأخيرة أدت إلى نتائج لم يكن من الممكن تصورها في السابق.

يصف التعلم العميق الخوارزميات التي تحلل البيانات بهيكل منطقي ، على غرار الطريقة التي يستنتجها الإنسان. لاحظ أن هذا يمكن أن يحدث من خلال التعلم الخاضع للإشراف وغير الخاضع للإشراف. لتحقيق هذا الهدف ، تستخدم تطبيقات التعلم العميق بُنية ذات طبقات (هرمية) من الخوارزميات تسمى الشبكات العصبية الاصطناعية. تصميم مثل هذه الشبكة العصبية الاصطناعية مستوحى من الشبكة العصبية البيولوجية للدماغ البشري ويؤدي إلى عملية تعلم أفضل بكثير من نماذج التعلم الآلي القياسية. من خلال التعلم العميق ، يمكن للآلات استخدام الصور أو النصوص أو الملفات الصوتية لتحديد وفعل أي شيء بطريقة تشبه الإنسان.

يركز التعلم العميق على طرق التعلم المتكررة التي تعرض الآلات لمجموعات البيانات الكبيرة. من خلال القيام بذلك ، فإنه يساعد أجهزة الكمبيوتر في العثور على خصائص من البيانات والتكيف مع التغييرات. يساعد التعرض المتكرر لمجموعات البيانات الآلات على فهم الاختلافات ومناطق البيانات والوصول إلى نتيجة موثوقة.

لماذا التعلم العميق مهم؟

يمكن أن ترتبط أهمية التعلم العميق بحقيقة أن عالمنا اليوم ينتج كميات هائلة من البيانات. والنتيجة هي الحاجة إلى نظام يمكنه هيكلة هذا الحجم الكبير من البيانات. يستخدم التعلم العميق هذا الحجم المتزايد من البيانات. يتم استخدام جميع المعلومات التي تم جمعها من هذه البيانات لتحقيق نتائج دقيقة من خلال نماذج التعلم العميق. يؤدي التحليل المتكرر لمجموعات البيانات الكبيرة إلى تقليل الأخطاء ، مما يؤدي في النهاية إلى نتيجة موثوقة. يستمر التعلم العميق في التأثير على الأعمال والمساحات الشخصية ويخلق العديد من فرص العمل في المستقبل.

على مدى السنوات القليلة المقبلة ، من المرجح أن تستخدم الشركات الناشئة وشركات التكنولوجيا الكبيرة والصغيرة التعلم العميق لترقية مجموعة واسعة من التطبيقات الحالية وإنشاء منتجات وخدمات جديدة (على الرغم من أن العديد من الشركات تستخدم بالفعل هذه التكنولوجيا الكبيرة). ستظهر خطوط وأسواق تجارية جديدة تمامًا ، والتي بدورها ستؤدي إلى

مزيد من الابتكار. سيكون استخدام أنظمة التعلم العميق أسهل وأكثر سهولة ، ومن المتوقع أن التعلم العميق سيغير بشكل أساسي طريقة تفاعل الناس مع التكنولوجيا ؛ مثلما غيّرت أنظمة التشغيل طريقة وصول الأشخاص العاديين إلى أجهزة الكمبيوتر.

كيف يعمل التعلم العميق؟

تم تصميم أنظمة التعلم العميق على غرار الشبكات العصبية في القشرة المخية الحديثة للدماغ البشري ، حيث يحدث الإدراك عالي المستوى. في الدماغ ، تنقل الخلايا العصبية معلومات كهربائية أو كيميائية. عند الاتصال بالخلايا العصبية الأخرى ، فإنها تشكل شبكة عصبية. في الآلات ، تتكون الخلايا العصبية بشكل مصطنع ، وإذا قمت بتوصيل ما يكفي من هذه الخلايا العصبية الاصطناعية ، تحصل على شبكة عصبية اصطناعية.

لكي تكون الشبكة العصبية مفيدة ، فإنها تحتاج إلى تدريب. لتدريب شبكة عصبية ، يتم إنشاء مجموعة من الخلايا العصبية الاصطناعية وتخصيص "وزن" رقمي عشوائي يحدد كيفية استجابة الخلايا العصبية للبيانات الجديدة. مثل أي طريقة إحصائية أو تعلم آلي ، ترى الآلة أولاً الإجابات الصحيحة. لذلك إذا لم تكتشف الشبكة بدقة ، على سبيل المثال ، عدم رؤية وجه في صورة ما ، فسيقوم النظام بضبط الأوزان (مقدار الاهتمام الذي يدفعه كل خلية عصبية للبيانات ، من أجل إنتاج الاستجابة الصحيحة). أخيرًا ، بعد التدريب المناسب ، ستكتشف الشبكة العصبية باستمرار الأنماط الصحيحة في الكلام أو الصور.

باختصار ، يعتمد جوهر التعلم العميق على طريقة تكرارية لتدريب الآلات لتقليد الذكاء البشري. تؤدي الشبكة العصبية الاصطناعية هذه الطريقة التكرارية من خلال عدة مستويات هرمية. تساعد المستويات الأساسية الآلات على تعلم معلومات بسيطة. كلما انتقلت إلى كل مستوى جديد ، تجمع الأجهزة المزيد من المعلومات وتدمجها مع ما تعلمته في المستوى الأخير. في نهاية العملية ، يجمع النظام جزءًا أخيرًا من المعلومات يمثل إدخالًا مختلطًا. تمر هذه المعلومات بعدة تسلسلات هرمية وتشبه التفكير المنطقي المعقد. دعنا نقسمها أكثر بمساعدة مثال. ضع في اعتبارك مساعدًا صوتيًا مثل Alexa أو Siri لترى كيف يستخدم التعلم العميق لتجارب المحادثة الطبيعية. في المراحل الأولى من الشبكة العصبية ، عندما يتغذى المساعد الصوتي على البيانات ، فإنه يحاول تحديد الأصوات والأشياء الأخرى. في المستويات العليا ، يلتقط معلومات المفردات ويضيف النتائج من المستويات السابقة. في المستويات التالية ، يحلل الإعلانات (الأوامر) ويجمع كل نتائجها. بالنسبة لأعلى مستوى من الهيكل الهرمي ، يتم تدريب المساعد الصوتي بدرجة كافية ليكون قادرًا على تحليل الحوار وتقديم المدخلات بناءً على تلك المدخلات.

تتيح العملية الهرمية لأنظمة التعلم العميق للآلات معالجة البيانات بطريقة غير خطية. تصنع الشبكات العصبية الاصطناعية مثل الدماغ البشرى ، والعقد العصبية متصلة مثل الشبكة. تتعلم الشبكات العصبية الاصطناعية شيئًا بسيطًا فى المستوى الأول من التسلسل الهرمى ثم تنتقل به إلى المستوى التالى. فى المستوى التالى ، يتم دمج هذه المعلومات البسيطة فى شىء أكثر تعقيدًا ويتم نقلها إلى المستوى التالى وهكذا. يبنى كل مستوى فى التسلسل الهرمى شيئًا أكثر تعقيدًا من المدخلات الواردة من المستوى السابق ، ويمكن أن يتعلم تلقائيًا استخراج السمات من مجموعات البيانات مثل الصور أو الفيديو أو النص ، بدون برمجة صريحة أو قواعد تقليدية.

فى التعلم العميق ، لا نحتاج إلى برمجة صريحة لكل شىء. يمكنهم التعرف تلقائيًا على تمثيلات البيانات مثل الصور أو الفيديو أو النص ، دون تقديم قواعد يدوية. يمكن أن تتعلم بنياتها عالية المرونة مباشرة من البيانات الخام ويمكن أن تزيد من أدائها التنبئى إذا تم توفير المزيد من البيانات.

كانت فكرة الخلايا العصبية الاصطناعية موجودة منذ 60 عامًا على الأقل، عندما طور فرانك روزنبيلتفي الخمسينيات من القرن الماضي "مدركًا" لكاشفات الضوء ودربه بنجاح على التمييز بين الأشكال الرئيسية. لكن الشبكات العصبية الأولية كانت محدودة للغايةفي عدد الخلايا العصبية التي يمكنها استنساخها. بمعنى أنهم لا يستطيعون التعرف على الأنماط المعقدة. ثلاث تطورات في العقد الماضي جعلت التعلم العميق أكثر شعبية واستدامة.

طور جيفري هينتون وباحثون آخرونفي جامعة تورنتو طريقة يمكن من خلالها للخلايا العصبية البرمجية أن تدرب نفسها عن طريق الطبقات. تتعلم الطبقة الأولى من الخلايا العصبية كيفية التعرف على الميزات الأساسية، مثل الحافة التي تحتوي على ملايين نقاط البيانات. عندما تتعلم إحدى الطبقات كيفية التعرف على هذه الأشياء بدقة، فإنها تنتقل إلى الطبقة التالية، والتي تدرب نفسها على تحديد ميزات أكثر تعقيدًا، مثل الأنف أو الأذنين. يتم بعد ذلك تغذية هذه الطبقة إلى طبقة أخرى، والتي تدرب نفسها على اكتشاف المزيد من مستويات التجريد، وهكذا، طبقة تلو الأخرى، حتى يتمكن النظام بشكل موثوق من اكتشاف الظواهر المعقدة جدًا مثل الإنسان.

التطور الثاني المسؤول عن التطورات الحديثةفي التعلم العميق هو الكمية الكبيرة من البيانات المتوفرة الآن. أدت الرقمنة السريعة إلى إنتاج بيانات واسعة النطاق، وهي الأكسجين المستخدم في تدريس أنظمة التعلم العميق.

أخيرًا، وجد فريقفي ستانفورد، بقيادة أندرو آنج، أن رقائق GPU، التي تم اختراعها للمعالجة البديهية لألعاب الفيديو، يمكن استخدامها للتعلم العميق. حتى وقت قريب، كان بإمكان رقائق الكمبيوتر التقليدية معالجة حدث واحد فقطفي كل مرة، ولكن تم تصميم وحدات معالجة الرسومات للحوسبة المتوازية. أدى استخدام هذه الشرائح لتنفيذ شبكات عصبية بملايين الاتصالات بالتوازي إلى تسريع تدريب وإمكانيات أنظمة التعلم العميق، مما سمح للآلة بالتعلم في يوم واحد شيئًا كان يستغرق عدة أسابيعفي السابق.

تتكون شبكات التعلم العميق الأكثر تقدمًا اليوم من ملايين الخلايا العصبية المحاكية مع مليارات الروابط فيما بينها ويمكن تدريبها من خلال التعلم غير الخاضع للإشراف. هذا هو التطبيق العملي الأكثر فعالية للذكاء الاصطناعي الذي تم اختراعه على الإطلاق. بالنسبة لبعض المهام، تكون أفضل أنظمة التعلم العميق للتعرف على الأنماط على قدم المساواة مع الأشخاص.

الفرق بين التعلم العميق والتعلم الآلي؟

خوارزميات التعلم العميق هي خوارزميات التعلم الآلي. ومن ثم ، قد تتساءل ما الذي يجعل التعلم الآلي العميق مميزًا. الإجابة: **بُنية خوارزميات الشبكة العصبية الاصطناعية ، والحاجة إلى تدخل بشري أقل ، والحاجة إلى بيانات أكبر.**

بادئ ذي بدء ، في حين أن خوارزميات التعلم الآلي التقليدية لها بُنية بسيطة نسبيًا ، مثل الانحدار الخطي أو شجرة القرار ، فإن التعلم العميق يعتمد على شبكة عصبية اصطناعية. تتكون هذه الشبكة العصبية الاصطناعية من عدة طبقات ، مثل الدماغ البشري ، فهي معقدة ومتشابكة. ثانيًا ، تتطلب خوارزميات التعلم العميق تدخلًا بشريًا أقل بكثير. على سبيل المثال ، إذا أردنا التعرف على علامة الإيقاف في صورة ما ، فإن خوارزمية تعلم الآلة التقليدية تتطلب من مهندس تحديد الخصائص والفئات يدويًا لفرز الصور والتحقق مما إذا كان الإخراج يطابق الحاجة. وإذا لم يكن الأمر كذلك ، فقد كان يعدل الخوارزمية. ومع ذلك ، باعتبارها خوارزمية التعلم العميق، يتم استخراج الميزات تلقائيًا وتتعلم الخوارزمية من أخطائها. ثالثًا ، يتطلب التعلم العميق بيانات أكثر بكثير من خوارزميات التعلم الآلي التقليدية لتعمل بشكل صحيح. يعمل التعلم الآلي مع آلاف من نقاط البيانات ، بينما يعمل التعلم العميق غالبًا بملايين النقاط. نظرًا لبنية العديد من الطبقات المعقدة ، يتطلب نظام التعلم العميق مجموعة كبيرة من البيانات للتخلص من التقلبات وإنشاء تفسيرات عالية الجودة.

وفقًا لذلك ، على الرغم من استخدام التعلم الآلي والتعلم العميق بالتبادل ، إلا أنهما ليسا شيئا واحداً. التعلم الآلي هو نطاق أوسع يستخدم البيانات لتحديد وإنشاء نماذج التعلم. يحاول التعلم الآلي فهم بُنية البيانات باستخدام النماذج الإحصائية ويبدأ بالتنقيب في البيانات. حيث تستخرج المعلومات ذات الصلة يدويًا من مجموعة البيانات ثم تستخدم الخوارزميات لتوجيه أجهزة الكمبيوتر للتعلم من البيانات وإجراء التنبؤات.

تم استخدام التعلم الآلي لفترة طويلة وتطور بمرور الوقت. التعلم العميق هو مجال جديد نسبيًا (ليس جديداً تمامًا ، حيث تطور التعلم العميق بمرور الوقت) يركز على التعلم والعمل فقط على الشبكات العصبية. في التعلم العميق ، كلما تم تغذية المزيد من البيانات إلى الشبكة ، زادت دقة النتائج ودقتها. يقودنا هذا إلى اختلاف آخر بين التعلم العميق والتعلم الآلي. بينما يمكن تحسين تعلم نماذج التعلم العميق بأحجام أكبر من البيانات ، فإن تعلم نماذج التعلم الآلي يقتصر على مستوى واحد. بمعنى آخر ، بعد مستوى معين ، يصلون إلى مستوى عالٍ من التعلم ، وأي إضافة

المزيد من البيانات الجديدة لا تحدث فرقًا. باختصار ، الاختلافات الرئيسية بين هذين المجالين هي كما يلي:

- **حجم مجموعة البيانات:** لا يعمل التعلم العميق بشكل جيد مع مجموعات البيانات الصغيرة. ومع ذلك ، يمكن لخوارزميات التعلم الآلي معالجة مجموعات البيانات الأصغر (أيضًا البيانات الكبيرة ولكن ليست كبيرة مثل مجموعات بيانات التعلم العميق) ، دون المساس بأدائها.
- **هندسة الميزات:** جوهر جميع خوارزميات التعلم الآلي هو هندسة الميزات وتعقيدها. في التعلم الآلي التقليدي ، يحدد الخبير الميزات التي يجب تطبيقها على النموذج. في التعلم العميق ، من ناحية أخرى ، يتم تنفيذ هندسة الميزات تلقائيًا من خلال الهيكل الهرمي للشبكات العميقة دون برمجة واضحة.
- **تبعيات الأجهزة:** يتطلب التعلم العميق أجهزة متقدمة لعمليات الحوسبة الثقيلة. من ناحية أخرى ، يمكن أيضًا تطبيق خوارزميات التعلم الآلي على الأجهزة المنخفضة الجودة. تتطلب خوارزميات التعلم العميق وحدات معالجة الرسومات لتحسين العمليات الحسابية المعقدة بشكل فعال.
- **وقت التنفيذ:** من السهل أن نتخيل أن خوارزمية التعلم الآلي سيكون لها وقت تنفيذ أقصر مقارنة بالتعلم العميق. لأن التعلم العميق يتطلب إطارًا زمنيًا أطول للتدريب ، ليس فقط بسبب مجموعة البيانات الضخمة ، ولكن أيضًا بسبب تعقيدات الشبكة العصبية. قد يستغرق تعلم خوارزمية التعلم الآلي من ثوانٍ إلى ساعات ، ولكن بالمقارنة، قد تستغرق خوارزميات التعلم العميق أسابيع.

نماذج التعلم العميق وحدها قادرة على إنشاء ميزات جديدة ، بينما في نهج التعلم الآلي، يجب تحديد الميزات بدقة من قبل المستخدمين.

فوائد التعلم العميق

عندما نفكر في التكنولوجيا، لا يمكن أن يكون هناك مناقشة للتعلم العميق. لاداعي للقول أن التعلم العميق أصبح أحد الجوانب الأكثر حيوية في التكنولوجيا. اليوم ، بالإضافة إلى الشركات والمؤسسات ، حتى الأشخاص في الجانب التكنولوجي يميلون إلى التعلم بعمق. أحد الأسباب العديدة التي تجعل التعلم العميق يجذب انتباه الجميع هو قدرته على عمل تنبؤات دقيقة. باختصار ، الشركات في وضع يمكنها من الاستفادة من مجموعة متنوعة من الفوائد المالية والتشغيلية من خلال التعلم العميق.

قد تسأل لماذا يستخدم عدد كبير من عمالقة التكنولوجيا التعلم العميق اليوم ، كما أن عدد هذه الشركات التي تستخدم التعلم العميق يتزايد يومًا بعد يوم. لفهم هذا السبب ، يجب على

المرء أن ينظر إلى الفوائد التي يمكن اكتسابها باستخدام نهج التعلم العميق. المدرجة أدناه هي بعض الفوائد الرئيسية لاستخدام هذه التكنولوجيا:

- **الاستخدام الأقصى للبيانات غير المهيكلة:** تُظهر الأبحاث أن نسبة كبيرة من بيانات المؤسسة غير مهيكلة ، لأن معظمها بصيغ مختلفة مثل الصور والنصوص وما إلى ذلك. بالنسبة لمعظم خوارزميات التعلم الآلي ، يعد تحليل البيانات غير المهيكلة أمرًا صعبًا. هذا هو المكان الذي يكون فيه التعلم العميق مفيدًا. لأنه يمكنك استخدام تنسيقات بيانات مختلفة لتعليم خوارزميات التعلم العميق وأيضًا اكتساب رؤى تتعلق بالغرض من التدريب. على سبيل المثال ، يمكنك استخدام خوارزميات التعلم العميق لاستكشاف العلاقات بين تحليل الصناعة ودردشة الوسائط الاجتماعية والمزيد للتنبؤ بأسعار الأسهم المستقبلية للمؤسسة.
- **لا حاجة لهندسة الميزات:** في التعلم الآلي ، تعد هندسة الميزات مهمة أساسية وهامة. هذا لأنه يحسن الدقة ، وفي بعض الأحيان قد تتطلب هذه العملية معرفة المجال حول مشكلة معينة. تتمثل إحدى أكبر مزايا استخدام نهج التعلم العميق في قدرته على أداء هندسة الميزات تلقائيًا. في هذا النهج ، تقوم الخوارزمية بمسح البيانات لتحديد الميزات ذات الصلة ثم دمجها لتسريع التعلم ، دون إخبارها صراحة. تساعد هذه الإمكانية علماء البيانات على توفير قدر كبير من الوقت ثم تحقيق نتائج أفضل.
- **تقديم نتائج عالية الجودة:** يصاب الناس بالجوع أو التعب ويرتكبون أخطاء أحيانًا. في المقابل ، ليس هذا هو الحال عندما يتعلق الأمر بالشبكات العصبية. يمكن لنموذج التعلم العميق المدرب بشكل صحيح أن ينجز آلاف المهام الروتينية والمتكررة في فترة زمنية قصيرة نسبيًا مقارنة بما يحتاجه الإنسان. بالإضافة إلى ذلك ، لن تنخفض جودة العمل أبدًا ، ما لم تحتوي بيانات التدريب على بيانات أولية لا تشير إلى مشكلة تريد حلها.

بالنظر إلى المزايا المذكورة أعلاه والاستفادة بشكل أكبر من نهج التعلم العميق، يمكن القول إن التأثير الكبير للتعلم العميق على التقنيات المتقدمة المختلفة مثل إنترنت الأشياءفي المستقبل واضح. لقد قطع التعلم العميق شوطًا طويلاً وأصبح سريعًا تقنية حيوية يتم استخدامها باستمرار من قبل مجموعة من الشركاتفي مجموعة متنوعة من الصناعات.

عيوب وتحديات التعلم العميق

على الرغم من تزايد أهمية التعلم العميق وتطوراته ، إلا أن هناك بعض الجوانب أو التحديات السلبية التي يجب معالجتها لتطوير نموذج التعلم العميق. مطلوب كميات كبيرة من البيانات لبدء تعلم خوارزمية التعلم العميق. على سبيل المثال ، يتطلب برنامج التعرف على الكلام بيانات متعددة اللهجات ، والتركيبة السكانية ، وبيانات ذات مقاييس زمنية للحصول على النتائج

المرجوة. في حين أن شركات مثل كوكل و مايكروسوفت قادرة على جمع الكثير من البيانات والحصول عليها ، فقد لا تتمكن الشركات الصغيرة ذات الأفكار الجيدة من القيام بذلك. أيضًا ، في بعض الأحيان ، قد لا تكون البيانات اللازمة لتدريب نموذج مبعثرة أو متوفرة.

على الرغم من أن نماذج التعلم العميق فعالة للغاية ويمكنها صياغة حل مناسب لمشكلة معينة بعد التدريب مع البيانات ، إلا أنها غير قادرة على القيام بذلك لحل مشكلة مماثلة وتحتاج إلى إعادة التدريب. لتوضيح ذلك ، ضع في اعتبارك خوارزمية التعلم العميق التي تتعلم أن الحافلات المدرسية دائمًا ما تكون صفراء ، ولكن فجأة تتحول الحافلات المدرسية إلى اللون الأزرق. ومن ثم ، يجب إعادة تدريسها. على العكس من ذلك ، ليس لدى الطفل البالغ من العمر خمس سنوات مشكلة في التعرف على السيارة كحافلة مدرسية زرقاء. بالإضافة إلى ذلك ، فهي أيضًا لا تعمل بشكل جيد في المواقف التي قد تكون مختلفة قليلاً عن البيئة التي مارسوا فيها. DeepMind، على سبيل المثال ، دربتها كوكل على هزيمة 49 لعبة Atari. ومع ذلك ، في كل مرة يهزم فيها النظام مباراة واحدة ، يجب إعادة تدريبه لهزيمة المباراة التالية. يقودنا هذا إلى قيد آخر للتعلم العميق ، وهو أنه في حين أن النموذج قد يكون جيداً للغاية في تعيين المدخلات إلى المخرجات، فقد لا يكون جيداً في فهم سياق البيانات التي يديرونها.

أخيرًا ، أكثر نقاط الضعف المعروفة في الشبكات العصبية هي طبيعة "الصندوق الأسود". ببساطة ، أنت لا تعرف كيف ولماذا حصلت شبكتك العصبية على ناتج معين. على سبيل المثال، عندما تقوم بإدخال صورة قطة إلى شبكة عصبية ويتنبأ بها الجهاز ، فمن الصعب جدًا فهم سبب هذا التوقع. سيكون هذا السيناريو مهمًا في قرارات العمل. هل يمكنك أن تتخيل أن الرئيس التنفيذي لشركة كبيرة يتخذ قرارًا بشأن ملايين الدولارات دون أن تفهم سبب قيامه بذلك؟ فقط لأن "الكمبيوتر" يقول أنه يجب أن يفعل ذلك؟ بالمقارنة ، الخوارزميات مثل أشجار القرار قابلة للتفسير بشكل كبير.

بشكل عام ، وفقًا لأندرو آنج ، يعد التعلم العميق طريقة رائعة "لبناء مجتمع قائم على الذكاء الاصطناعي" ، والتغلب على أوجه القصور هذه بمساعدة التقنيات الأخرى هو الطريق الصحيح للوصول الى الهدف.

لا نستخدم التعلم الآلى على الرغم من وجود التعلم العميق؟

الجواب لا. هذا لأن التعلم العميق يمكن أن يكون مكلفًا من الناحية الحسابية. إذا كان من الممكن حل مشكلة ما باستخدام خوارزمية تعلم آلي أبسط ، أي خوارزمية لا تحتاج إلى التعامل مع مجموعة معقدة من الميزات الهرمية في البيانات ، فإن خيارات حسابية أقل هي الخيار الأفضل. لذلك ، قد تكون خوارزميات التعلم الآلي مرغوبة أكثر إذا كنت بحاجة إلى نتائج أسرع. فهي أسرع في التدريب وتتطلب طاقة حوسبية أقل.

قد لا يكون التعلم العميق أيضًا الخيار الأفضل للتنبؤ المستند إلى البيانات. على سبيل المثال، إذا كانت مجموعة البيانات صغيرة ، فقد تؤدي أحيانًا نماذج تعلم الآلة الخطية الأبسط إلى نتائج أكثر دقة. ومع ذلك ، يجادل بعض خبراء التعلم الآلي بأن شبكة عصبية عميقة جيدة التدريب لا يزال بإمكانها العمل بشكل جيد مع كميات صغيرة من البيانات.

وفقًا لذلك ، متى يجب استخدام التعلم الآلي أو التعلم العميق؟ ذلك يعتمد على الاحتياجات الخاصة بك:

- **هل تحتاج إلى نموذج دقيق للغاية؟** استخدم التعلم العميق.
- **هل تحتاج إلى نموذج خفيف الوزن؟** استخدم التعلم الآلي.
- **هل تعمل على مشكلة في رؤية الكمبيوتر؟** أصبحت شبكات التعلم العميق ، مثل الشبكات الالتفافية ، قياسية الآن في هذا السيناريو.
- **لست متأكداً من احتياجاتك؟** حاول حل المشكلة عن طريق التعلم الآلي أولاً ، إذا كانت النتائج غير مرضية بدرجة كافية ، فاختبر التعلم العميق.

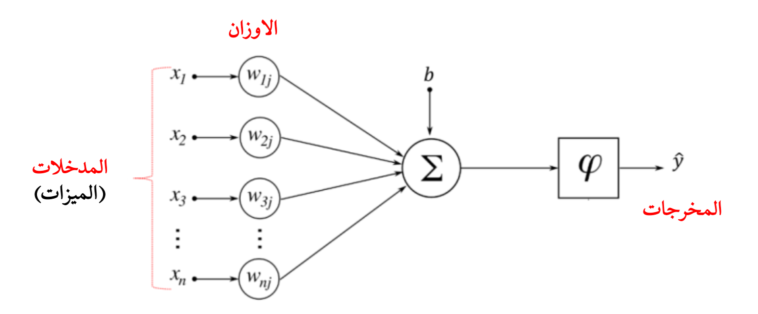
الشبكة العصبية

الخلايا العصبية الاصطناعية

الخلايا العصبية الاصطناعية (العصبونات) هي اللبنات الرئيسية للشبكات العصبية الاصطناعية. يصفون نموذجًا رياضيًا بسيطًا مستوحى من الخلايا العصبيةفي الدماغ. تتمثل الوظيفة الأساسية للخلايا العصبية الاصطناعيةفي تلقي مدخلات متعددة $x_1, \ldots, x_n$ وحساب مجموع الاوزان z لهذه المدخلات باستخدام الأوزان $w_1, \ldots, w_n$. مجموع الاوزان z هو تحويل خطي لمدخلات الخلايا العصبية. بالإضافة إلى ذلك، يضاف التحيز b إلى مجموع الاوزان للمدخلات ويتم تمرير النتيجة من خلال دالة التنشيط φ ، مما ينتج عنه إخراج نهائي لـ $\hat{y}$:

$$\hat{y} = \varphi(b + \sum_{i=1}^{n} w_i \times x_i)$$

الشكل 7ـ1 عبارة عن مخطط عالي المستوى يوضح العلاقة بين متجه الإدخال ومتغير الإخراج. قيم المعاملات w و b غير معروفة. ومن ثم ، في بداية عملية التعلم ، يتم تعديل هذه القيم حسب الرغبة. ثم يقوم نظام التعلم الآلي بتحسينها باستخدام خوارزميات التحسين. ومن ثم ، يمكن تدريب الخلايا العصبية الاصطناعية لتقريب دالة فيما يتعلق بمدخلات x.



الشكل 7ـ1. الخلية العصبية الصناعية.

على الرغم من أن هذا نموذج بسيط للغاية ، يمكن استخدام الخلايا العصبية الاصطناعية لأغراض متنوعة. إذا تم تحديد الدالة الخطوية كدالة التنشيط ، يمكن استخدام الخلايا العصبية للتصنيف الثنائي. يُطلق على هذا النوع من الخلايا العصبية المستخدمة في التصنيف الثنائي اسم **بيرسيبترون Perceptron**. ومع ذلك ، فإن خوارزمية بيرسيبترون يمكنها فقط تصنيف مدخلاتها بشكل صحيح عندما تكون البيانات قابلة للفصل خطيًا حسب الفئة. لحل هذه المشكلة ، تم اقتراح "بيرسيبترون متعدد الطبقات" ، والذي يمكن رؤيته على أنه سلسلة من بيرسيبترون منظم في "طبقات" ، كل منها يتلقى مدخلاته من البيرسيبترون السابق.

الشبكة العصبية امامية التغذية

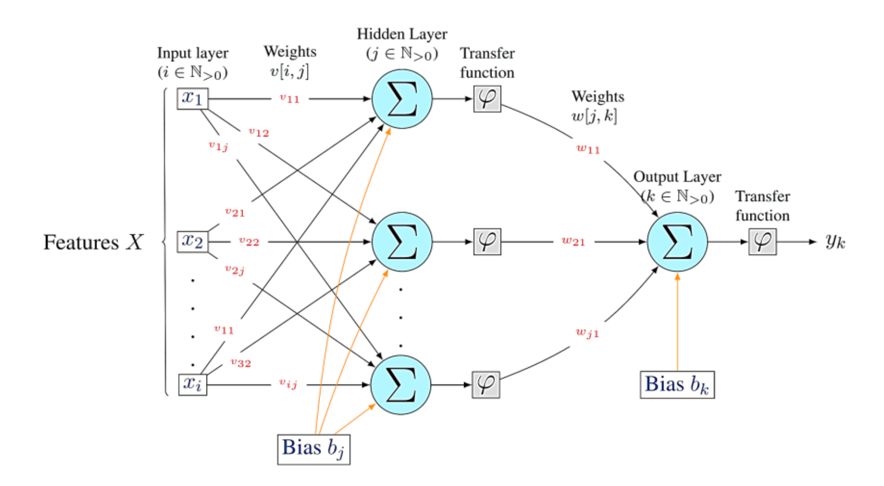
تشبه الشبكة العصبية امامية التغذية الخلايا العصبية الاصطناعية التي تمت مناقشتها سابقًا. والغرض منه أيضًا هو تقريب دالة مدخلات x. ومع ذلك ، بدلاً من الاقتصار على دوال بسيطة للغاية تتكون من مجموعة الاوزان فقط مع دالة التنشيط ، تجمع الشبكات العصبية امامية التغذية بين عدة عصبونات لتشكيل رسم بياني موجه. يظهر مثال للشبكة العصبية امامية التغذية في الشكل 7ـ2. تتكون كل شبكة عصبية امامية التغذية من **مدخلات** (يشار إليها عادة باسم **طبقة الإدخال**) ، وعدد عشوائي من الطبقات الوسيطة للخلايا العصبية تسمى **الطبقات المخفية**[1]، وطبقة تحسب المخرجات تسمى **طبقة المخرجات**. هذا النهج القائم على الطبقة هو المكان الذي يسمى التعلم العميق ، لأن عمق الشبكة العصبية امامية التغذية يصف عدد الطبقات التي تشكل الشبكة العصبية امامية التغذية. عندما تتصل كل خلية عصبية في طبقة ما بجميع الخلايا العصبية في الطبقة التالية ، فإنها تسمى شبكة **متصلة بالكامل**[2] ؛ تسمى الطبقات التي تظهر هذا السلوك **طبقات متصلة بالكامل**. شبكة العصبية امامية التغذية البسيطة الموضحة في الشكل 7ـ2 هي الآن أقوى بكثير من الخلايا العصبية الاصطناعية. يمكن إثبات أنه يمكن استخدام الشبكات

---

[1] hidden layers

[2] fully-connected network

العصبية أمامية التغذية ذات طبقة مخفية واحدة لتقريب أي دالة مستمرة. تمثل الشبكات العصبية امامية التغذية أساس معظم تطبيقات التعلم العميق. على سبيل المثال ، **الشبكات العصبية الالتفافية**[1] ذات الشعبية الكبيرة والناجحة هي مجرد امتدادات للشبكات العصبية امامية المغذية القياسية.



شكل 7_2. البُنية البسيطة لشبكة العصبية امامية التغذية

تحسين الشبكة العصبية

الغرض من تحسين الشبكات العصبية امامية التغذية هو العثور تلقائيًا على الأوزان والتحيزات التي تقارب إخراج الشبكة المستهدفة y مع الإدخال x. من أجل تحقيق هذا الهدف ، من الضروري تحديد مقياس لمقدار تقدير الشبكة العصبية امامية التغذية. يشار إلى هذا المعيار عادة باسم **دالة الخسارة**[2] **أو دالة التكلفة**[3] $J(\theta)$ ، حيث تصف معاملات الشبكة (الأوزان والتحيزات). إعطاء مجموعة من N العينات التدريبية $x_T = [x_{T_1}, x_{T_2}, \dots, x_{T_N}]$ والعلامة ذات الصلة $y = [y_1, y_2, \dots, y_N]$، عادةً ما يتم حساب $J(\theta)$ على أنه متوسط دالة الخسارة لكل عينة $L(\hat{y}(x_{T_i}; \theta), y_i)$ حيث $\hat{y}(x_{T_i}; \theta)$ ناتج شبكة العصبية امامية التغذية ، وفقاً للتدريب عينة x_{T_i} كمدخلات ومعاملات الشبكة θ:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(\hat{y}(x_{T_i}; \theta), y_i)$$

---

[1] Convolutional Neural Network

[2] loss function

[3] cost function

عادةً ما تكون قيمة دالة الخسارة الأصغر مساوية لتقريب أفضل دالة للشبكة العصبية امامية التغذية، لذلك يمكن صياغة طريقة التدريب للشبكة العصبية امامية التغذية كمشكلة تحسين ، حيث يكون الهدف هو تقليل دالة الخسارة $J(\theta)$ وفقًا لـ معاملات الشبكة هي θ. يتم ذلك عادةً باستخدام نوع من الخوارزمية يسمى الانحدار الاشتقاقي.

الانحدار الاشتقاقي[1]

بالنظر إلى دالة الخسارة ذات القيم الحقيقية لـ $J(\theta)$ ، فإن الغرض من الانحدار الاشتقاقي (التدرج المتناقص) هو إيجاد الحد الأدنى المحلي $J(\theta)$ فيما يتعلق بالمعاملات θ. في حين أنه قد يكون من الممكن تحليليًا حساب الحد الأدنى من الخسارة لدوال الخسارة البسيطة ، إلا أنه من المستحيل بالنسبة للدوال الأكثر تعقيدًا ذات المعاملات المتعددة ، مثل دوال الخسارة لشبكات العصبية امامية التغذية بملايين من المعاملات. على عكس الحسابات التحليلية للحدود الدنيا ، فإن الانحدار الاشتقاقي هو نهج رقمي يبدأ باختيار معاملات عشوائية ويتحرك بشكل متكرر في الاتجاه السلبي لتدرج الدالة للعثور على الحد الأدنى المحلي. بالنسبة لعينة تدريبية واحدة x_{T_i} مع الناتج المستهدف المقابل y_i، يتم حساب الانحدار الاشتقاقي على النحو التالي:

$$-g_\theta = -\nabla_\theta L\left(\hat{y}\left(x_{T_i};\theta\right), y_i\right)$$

يتم الحصول بعد ذلك على الانحدار النهائي لدالة الخسارة $J(\theta)$ عن طريق حساب متوسط جميع التدرجات في مجموعة التدريب x_T بأكملها:

$$-g_\theta = -\frac{1}{N}\sum_{i=1}^{N} - g_{\theta_i}$$

من خلال تحديد عامل إيجابي يتحكم في حجم الانحدار الاشتقاقي ، يسمى معدل **التعلم**[2] η ، يمكن تعريف قانون الانحدار الاشتقاقي بتحديث معاملات الشبكة العصبية امامية التغذية على أنه:

$$\theta = \theta + \eta - g_\theta$$

اعتمادًا على القيمة الأولية للمعاملات ، من الممكن أن يجد الانحدار الاشتقاقي الحد الأدنى العالمي $J(\theta)$ ، لكن هذا غير مضمون ما لم يكن $J(\theta)$ محدبًا. النقطة المهمة هي أنه لكي يتقارب الانحدار الاشتقاقي ، يجب أن تكون دالة الخسارة سلسة وتوفر تدرجات في كل مكان. هذا هو السبب في أن دالة الخسارة الانتقائية في بعض الأحيان تختلف عادةً عن الغرض الفعلي ويتم استخدام دالة خسارة بديلة بدلاً من ذلك. على سبيل المثال ، بدلاً من تحسين العدد الصحيح

---

[1] gradient descent

[2] learning rate

لعينات التصنيف في مشكلة تصنيف الصورة ، يمكننا تحسين الموثوقية المتوقعة وضمان الهدف لكل فئة إلى متوسط الخطأ التربيعي (MSE). في حين أن عدد العينات المصنفة جيداً سيكون منفصلًا وبالتالي غير متساوٍ ، سيكون MSE في كل مكان ويوفر تدرجات مفيدة لتقليل التدرجات. على سبيل المثال ، غالبًا ما يتم استخدام الاحتمال اللوغاريتمي السلبي لفئة حقيقية كبديل للخسارة 0ــ1. يسمح اللوغاريتم السلبي للنموذج بتقدير الاحتمال الشرطي للفئات فيما يتعلق بالمدخلات ، وإذا كان النموذج قادرًا على القيام بذلك بشكل جيد ، فيمكنه تحديد الفئات التي تسفر عن أقل خطأ تصنيف في التوقع الرياضي.

الحساب الكفوء للانحدار الاشتقاقي في الانتشار الخلفي

ربما تكون خوارزمية الانتشار الخلفي هي اللبنة الأساسية في الشبكة العصبية. يعد الانتشار الخلفي في الأساس طريقة ذكية لحساب التدرجات بشكل فعال في الشبكات العصبية متعددة الطبقات. بمعنى آخر ، يتعلق الانتشار الخلفي بحساب الانحدار الاشتقاقي للدوال المتداخلة ، والذي يتم تمثيله كرسم بياني حسابي باستخدام قاعدة السلسلة. ببساطة ، بعد كل تمريرة امامية عبر شبكة، يقوم الانتشار الخلفي بتمريرة للخلف أثناء ضبط معاملات النموذج (الأوزان والتحيزات).

قبل وصف الخوارزمية ، دعنا نصف رمزًا قائمًا على مصفوفة لفهم أفضل. يصف الوزن w_{jk}^{l} الوزن الذي يربط الخلايا العصبية kام في الطبقة $l-1$ بالخلايا العصبية j في الطبقة l. من هذه الأوزان ، يمكن تكوين مصفوفة w^{l}، والتي تسمى **مصفوفة الوزن** للطبقة lام، حيث يكون مدخل الصف jام والعمود kام مساويًا لـ w_{jk}^{l}. وفقًا للخلايا العصبية K في الطبقة $l-1$ والخلايا العصبية J في الطبقة l ، تكون هذه المصفوفة على النحو التالي:

$$w^{l}=\begin{bmatrix} w_{11}^{l} & \cdots & w_{1K}^{l} \\ \vdots & \ddots & \vdots \\ w_{J1}^{l} & \cdots & w_{JK}^{l} \end{bmatrix}$$

وبالمثل ، يمكن توجيه تحيزات b_{j}^{l} ومخرجات **التنشيط المسبق**[1] لـ z_{j}^{l}ومخرجات y_{j}^{l} من الخلايا العصبية في الطبقة l كـ b^{l}، z^{l} و y^{l} ، على التوالي:

$$b^{l}=\begin{bmatrix} b_{1}^{l} \\ \vdots \\ b_{J}^{l} \end{bmatrix},\qquad z^{l}=\begin{bmatrix} z_{1}^{l}=\sum_{k} w_{jk}^{l} y_{k}^{(l-1)}+b_{j}^{l} \\ \vdots \\ z_{J}^{l}=\sum_{k} w_{jk}^{l} y_{k}^{(l-1)}+b_{j}^{l} \end{bmatrix},\qquad y^{l}=\begin{bmatrix} y_{1}^{l}=\varphi(z_{1}^{l}) \\ \vdots \\ y_{J}^{l}=\varphi(z_{J}^{l}) \end{bmatrix}$$

كخطوة أولى في خوارزمية الانتشار الخلفي ، يتم حساب نتائج **الانتشار الخلفي**. خلال هذه الخطوة ، تتلقى الشبكة العصبية للتغذية عينة تدريب واحدة x_{T_i} كمدخلات وتحسب مخرجات

---

[1] pre-activation

كل طبقة y^l بالإضافة إلى مخرجات التنشيط المسبق لكل طبقة z^l إلى الطبقة النهائية D. بالإضافة إلى ذلك ، وفقًا للإخراج النهائي لشبكة التغذية العصبية النهائية y^D ، والإخراج المستهدف y_i لعينة التدريب هذه ، يتم حساب خطأ كل عينة $L(y^D, y_i)$. يسمى هذا بالانتشار الامامي لأنه يتم حساب مخرجات الشبكة بشكل تكراري لكل طبقة ، من طبقة الإدخال إلى طبقة الإخراج. بعد ذلك ، يتم تنفيذ الانتشار الخلفي للتدرج عن طريق حساب التدرج g_y^D في طبقة الإخراج D فيما يتعلق بإخراج الشبكة العصبية للتغذية:

$$g_y^D = \nabla_{y^D} L(y^D, y_i)$$

نظرًا لأننا مهتمون بالتدرج فيما يتعلق بإخراج التنشيط المسبق ، فإننا نحسبه عن طريق إجراء عملية ضرب هادامارد[1] ⊙ بين تدرج التنشيط اللاحق[2] ومشتق دالة التنشيط:

$$g_z^D = g_y^D \odot \varphi'(z^D)$$

ثم يمكن استخدام تدرج التنشيط السابق هذا g_z^D لحساب تدرجات دالة التكلفة $J(\theta)$ فيما يتعلق بمصفوفة الوزن W^D ومتجه التحيز b^D:

$$\nabla_{b^D} L(y^D, y_i) = g_z^D$$

$$\nabla_{W^D} L(y^D, y_i) = g_z^D (y^{(D-1)})^T$$

للوصول إلى الانحدار الاشتقاقي لتنشيط المسبق الشبكة للطبقة السفلية التالية $(D-1)$ ، نقوم ببساطة بضرب مصفوفة التحويل $(W^D)^T$ بواسطة تدرج التنشيط المسبق للطبقة الحالية g_z^D:

$$g_y^{(D-1)} = \nabla_{z^{D-1}} L(y^D, y_i) = (W^D)^T g_z^D$$

يمكن الآن تكرار الخطوات المذكورة أعلاه عدة مرات حتى يتم تحديد كل تدرج فردي ، وفقًا لكل وزن وتحيز ، في الطبقة الأولى ، مما يؤدي إلى التدرج النهائي لكل عينة.

$$g_{\theta_i} = \nabla_\theta L(\hat{y}(x_{T_i}; \theta), y_i)$$

من الأمثلة التدريبية x_{T_i} يصبح الناتج المستهدف y_i. ومن ثم ، باستخدام خوارزمية الانتشار الامامي ، أصبح لدينا الآن طريقة لحساب الانحدار الاشتقاقي لعينة تدريبية واحدة في المعادلة.

$$g_{\theta_i} = \nabla_\theta L(\hat{y}(x_{T_i}; \theta), y_i)$$

وهو أمر ضروري لتحديث معاملات الشبكة العصبية للتغذية الامامية بسبب الانحدار الاشتقاقي.

---

[1] Hadamard product

[2] post-activation

ومع ذلك ، فإن العيب الرئيسي في تحسين الشبكات العصبية من خلال الانتشار الخلفي والانحدار الاشتقاقي هو مشكلة **تلاشي الانحدار**[1]. كما هو موضح أعلاه ، يتم حساب الانحدار الاشتقاقي لوزن معين عن طريق الانتشار الامامي من الإخراج إلى الخلايا العصبية المقابلة. في مسار الانتشار الامامي هذا ، يتم ضرب التدرجات بشكل متكرر من الخلف إلى الأمام باستخدام قاعدة السلسلة في حساب التفاضل والتكامل. وبالتالي ، عندما تكون التدرجات صغيرة (والتي يمكن أن تحدث ، على سبيل المثال ، عند استخدام دوال التنشيط المشبعة[2] Sigmoid) ، يمكن أن يؤدي ذلك إلى تدرجات صغيرة للخلايا العصبية في الطبقات الأمامية للشبكة ، مما يؤدي إلى بطء التعلم لهذه الخلايا العصبية.

في حين أن هذه مشكلة في مرحلة الانتشار الخلفي وتلاشي الانحدار ، إلا أن الأساليب الحديثة للتعلم العميق تعالج هذه المشكلة بشكل أكبر. لا يشبع المنشط ReLU حجم التدرج في منطقة الإدخال الإيجابية. نتيجة لذلك ، لا تواجه تلاشي الانحدار في تلك المنطقة. بالإضافة إلى ذلك، هناك نوع جديد من بنية الشبكة يسمى ResNet يتجاوز الخلايا العصبية أو الطبقات ذات التدرجات المتلاشية باستخدام مسارات مختصرة ، مما يسمح للتدرجات بالتدفق عبر شبكات أعمق بكثير دون تجاوز. من الممكن أيضًا تثبيت توزيع المدخلات عن طريق تسوية الطبقات الوسطى باستخدام **التسوية الجماعية**[3] ، وبالتالي تقليل احتمالية التعثر في الحالات المشبعة.

الانحدار الاشتقاقي العشوائي

في حين أن الانحدار الاشتقاقي القياسي باستخدام الانتشار الخلفي هو طريقة مفيدة للتدريب الآلي للشبكات العصبية الأمامية ، عندما تكون مجموعة التدريب كبيرة جدًا ، يتطلب الانحدار الاشتقاقي الكثير من الموارد الحسابية. لأن خطوة تحديث واحدة فقط تتطلب حساب جميع التدرجات لجميع عينات التدريب.

الغرض من **الانحدار الاشتقاقي العشوائي** [4] (SGD) هو تسريع عملية التعلم مع تغيير طفيف في إجراء الانحدار الاشتقاقي القياسي. يتمثل الاختلاف الرئيسي بين SGD والانحدار الاشتقاقي القياسي في أن SGD تحسب تدرج دالة الخطأ $\nabla_\theta j(\theta)$ عن طريق حساب تدرجات الخطأ لكل عينة فقط لمجموعة فرعية صغيرة من m عينة تدريبية مختارة عشوائيًا من مجموعة التدريب $x_M = [x_{M_1}, \dots, x_{M_m}] \subset x_T$، مع الناتج المستهدف المقابل $y_M = [y_{M_1}, \dots, y_{M_m}] \subset$. تسمى هذه المجموعة الفرعية من العينات التدريبية **بالدفعة الصغيرة**[5]، m هو عدد حالات هذه

---

[1] vanishing gradient problem

[2] saturating

[3] batch normalization

[4] Stochastic Gradient Descent

[5] minibatch

الدفعات الصغيرة ، والتي تسمى **حجم الدفعة الصغيرة**. يمكن تلخيص تقدير التدرج باستخدام الدفعات الصغيرة:

$$\nabla_\theta j(\theta) = \nabla_\theta \frac{1}{N}\sum_{i=1}^{N} L\left(\hat{y}\left(x_{T_i};\theta\right), y_i\right) \approx \nabla_\theta \frac{1}{m}\sum_{i=1}^{m} L\left(\hat{y}\left(x_{M_i};\theta\right), y_{M_i}\right)$$

الانحدار الاشتقاقي مع الزخم[1]

كما ذكرنا سابقًا ، يمكن استخدام الانحدارات الاشتقاقية لإيجاد الحد الأدنى المحلي للدالة. ومع ذلك ، اعتمادًا على شكل الدالة ، غالبًا ما يؤدي النهج التكراري للانحدار الاشتقاقي إلى عدد كبير من الخطوات ؛ الانحدار الاشتقاقي بطيء ، خاصة بالنسبة للدوال التي تتضمن العديد من المناطق شبه المسطحة ذات المنحدرات الصغيرة. لحل هذه المشكلة ، يُضاف **الزخم** إلى خوارزمية الانحدار الاشتقاقي. الفكرة الرئيسية للزخم هي إضافة ذاكرة قصيرة المدى إلى الانحدار الاشتقاقي ، والذي يسمى أحيانًا **التسريع**[2]. وفقًا لذلك ، يتم تغيير خطوة تحديث الوزن على النحو التالي:

$$g_\beta = \beta g_\beta + \frac{1}{N}\sum_{i=1}^{N} g_{\theta_i}$$

$$\theta = \theta - \eta g_\beta$$

عندما تكون g_β الأولية صفرًا ، η معدل التعلم و β معايير الزخم.

تحسين معدل التعلم التكيفي

على الرغم من أن SGD هي طريقة تحسين قوية للغاية لتدريب الشبكات العصبية ، إلا أنه لا يزال من المهم اختيار أفضل معدل تعليمي η لكل مسئلة. إذا تم اختيار η كبير جدًا ، فقد يتقلب التدريب أو لا يتقارب أو يتجاوز الحدود الدنيا المحلية ذات الصلة. في المقابل ، إذا تم اختيار معدل تعليمي صغير جدًا ، فإنه يؤخر بشكل كبير عملية التقارب. ومن ثم ، فإن الأسلوب الشائع للتحايل على هذا هو استخدام معدل **اضمحلال التعلم**[3]. على سبيل المثال ، باستخدام الانحدار الاشتقاقي ، يمكن تقليل معدل التعلم إلى حد ما على مدار عدة فترات. هذا يسمح بقدر أكبر من التعلم في بداية التدريب ومعدل تعلم أقل في نهاية التدريب. ومع ذلك ، فإن طريقة الانحلال هذه هي أيضًا معامل فائق في حد ذاتها ويجب تصميمها بعناية اعتمادًا على التطبيق.

---

[1] Momentum

[2] acceleration

[3] rate decay

الهدف من مُحسِّن معدل التعلم التكيفي هو حل مشكلة إيجاد معدل التعلم الصحيح. في هذه الأساليب ، لا يعد معدل التعلم متغيرًا شاملاً ، ولكن بدلاً من ذلك يكون لكل معامل قابلة للتعليم معدل التعلم الخاص بها. في حين أن هذه الطرق لا تزال بحاجة إلى معاملات ، فإن الحجة الرئيسية هي أنها تعمل بشكل جيد مع نطاق أوسع من التكوينات ؛ في كثير من الأحيان عندما يستخدمون فقط المعاملات الفائقة الافتراضية المقترحة.

التدابير

جنبًا إلى جنب مع تقنية الانحدار الاشتقاقي ، يوفر الانتشار الخلفي طريقة فعالة لتحسين معاملات تدريب الشبكة العصبية لتقليل دالة خطأ معينة. ومع ذلك ، فإن تدريب الشبكات العصبية في الممارسة العملية يعوقه عدد من العوامل. أولاً ، يؤدي الجمع بين العديد من طبقات المعالجة غير الخطية في الشبكة العصبية إلى جعل دالة الخطأ الناتجة غير محدبة للغاية. يعني غير محدب أنه عند تقليل دالة الخطأ للشبكة عن طريق أساليب الانحدار الاشتقاقي ، لا يوجد ضمان بأن نقطة ثابتة معينة هي الحد الأدنى العالمي. أيضًا ، قد يكون للطريقة التي يتم بها تهيئة المعاملات قبل التحسين وتقنية الانحدار الاشتقاقي المستخدمة تأثير قوي جداً على أفضل حل تم العثور عليه أثناء التحسين. أيضًا ، الشبكات العصبية امامية التغذية متعددة الطبقات عرضة لمشكلة تلاشي الانحدار والانفجار. لذلك ، هناك حاجة لاتخاذ تدابير لمنع هذه المشاكل.

تهيئة الاوزان الاولية

في الشبكات العصبية ، يجب اختيار القيمة الأولية للأوزان بعناية فائقة. على سبيل المثال، إذا كان للعديد من الخلايا العصبية في طبقة مخفية نفس الأوزان ، فسوف تتلقى نفس التدرجات. ومن ثم ، فإنهم يحسبون نفس النتائج ، مما يؤدي إلى إهدار سعة النموذج. عادة ، يتم تهيئة أوزان الشبكة العصبية باستخدام توزيع غاوسي بمتوسط صفر وانحراف معياري صغير. ومع ذلك ، فإن المشكلة تكمن في أن توزيع ناتج خلية عصبية تمت تهيئتها عشوائيًا له تباين يزيد مع عدد المدخلات. لتسوية تباين الإخراج لكل خلية عصبية إلى 1، ما عليك سوى استخدام التوزيع الطبيعي القياسي وقياس الوزن بناءً على الجذر التربيعي لسعة الإدخال[1] n_{in} ، وهو عدد المدخلات:

$$w_0 \sim \frac{\mathcal{N}(0,1)}{\sqrt{n_{in}}}$$

وبالمثل ، قام Gloret و Benjiou بتحليل الانحدارات الاشتقاقية للانتشار الخلفي وأوصيا بالتهيئة (المعروفة باسم تهيئة القيم الاولية Javier أو Gloret).:

[1] fan-in

$$w_0 \sim \sqrt{\frac{2}{n_{in}+n_{out}}}\mathcal{N}(0,1)$$

حيث يصف n_{out} عدد وحدات الإخراج. تم تقديم المواصفات خصيصًا للخلايا العصبية مع تنشيط ReLU:

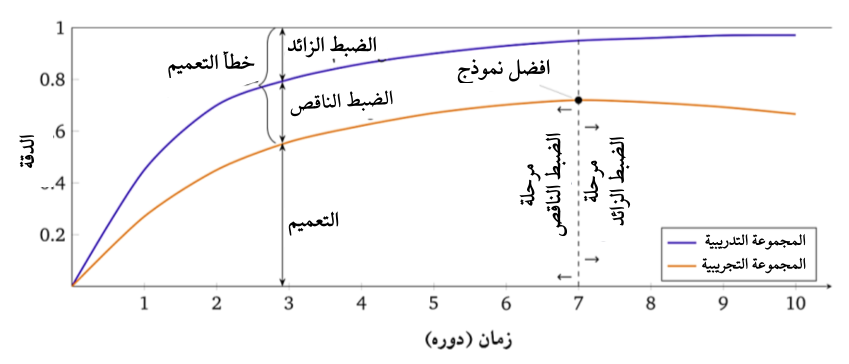
$$w_0 \sim \sqrt{\frac{2}{n_{in}}}\mathcal{N}(0,1)$$

التنظيم

حتى الآن ، أشرنا فقط إلى تدريب شبكة تغذية عصبية بالانحدار الاشتقاقي والانتشار الخلفي باستخدام مجموعة تدريب x_T مع علامات y المقابلة. بينما يمكننا بهذه الطريقة تدريب شبكتنا العصبية على التنبؤ بمخرجات مجموعة التدريب ، إلا أن هذا لا يعني بالضرورة أنها قادرة على التنبؤ بدقة بمخرجات البيانات غير المرئية. لذلك ، كما هو مذكور في الفصل 5 ، يتم تقديم مجموعتين إضافيتين من البيانات للتحسين ، مجموعة التحقق من الصحة والمجموعة التجريبية. كل مجموعات البيانات الثلاث مستقلة عن بعضها البعض ، لذلك لا توجد حالات مشتركة بينها.

تُستخدم مجموعات التحقق من الصحة في الشبكات العصبية بشكل شائع لضبط المعاملات الفائقة للنموذج مثل بُنية الشبكة أو معدلات التعلم. تستخدم مجموعة الاختبار فقط للتقييم النهائي من أجل تقييم أداء الشبكة في البيانات غير المرئية. إذا لم تكن الشبكة العصبية معممة بشكل جيد ، أي أن خطأ التدريب أقل من خطأ الاختبار ، كما ذكرنا سابقًا ، تسمى هذه الحالة الضبط الزائد Overfitting. من ناحية أخرى ، يُطلق على السيناريو العكسي الاستدارة عندما تكون خطأ الاختبار أقل بكثير من خطأ التدريب ، الضبط الناقص (الشكل 7ــ3). عادةً ما يرتبط الضبط الزائد والضبط الناقص في الشبكات العصبية العميقة ارتباطًا مباشرًا بقدرة النموذج. ترتبط سعة النموذج لشبكة عصبية عميقة ارتباطًا مباشرًا بعدد المعاملات داخل الشبكة. تحدد سعة النموذج مدى قدرة الشبكة العميقة على ملاءمة مجموعة واسعة من الدوال. إذا كانت السعة منخفضة جداً ، فقد لا تتمكن الشبكة من تكييف مجموعة التدريب (الضبط الناقص) ، بينما قد تؤدي سعة النموذج الكبيرة جداً إلى الاحتفاظ بعينات التدريب (الضبط الزائد). عادة لا يمثل الضبط الناقص مشكلة بالنسبة للشبكات العصبية العميقة. هذا لأنه يمكن حل هذه المشكلة باستخدام بُنية شبكة أقوى أو أعمق مع المزيد من المعاملات. ومع ذلك ، من أجل التمكن من استخدام الشبكات العميقة للبيانات الجديدة وغير المرئية ، يجب السيطرة على الضبط الزائد.

إن عملية تقليل تأثير الضبط الزائد أو منعه (ليس تماما. لماذا؟[1]) تسمى التنظيم. في هذا القسم ، نصف بإيجاز الأساليب الأكثر شيوعًا للتنظيم في الشبكات العميقة.



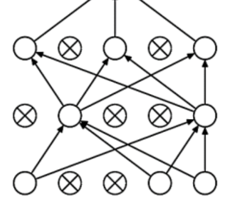
الشكل 7ـ3. سلوك التعميمفي منحنى التعلم وفق معيار الدقةفي البيانات التدريبية والتجريبية

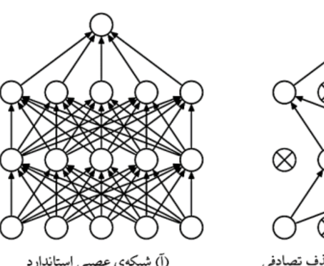
التوقف المبكر

عندما تكون سعة النموذج لشبكة عميقة كبيرة بما يكفي لتكون قادرة على الضبط الزائد ، فمن الشائع ملاحظة أن الخطأ التدريبي يتناقص بشكل طردي حتى التقارب ، في حين أن خطأ التحقق من الصحة يتناقص في البداية ويزداد مرة أخرى بعد فترة. الغرض من التوقف مبكرًا هو تنظيم الشبكة العميقة من خلال إيجاد معاملات الشبكة عند النقطة بأقل خطأ للتحقق من الصحة. باستخدام معاملات الشبكة مع أقل خطأ للتحقق من الصحة ، من المحتمل أن تكون الشبكة معممة بشكل أفضل للبيانات غير المرئية. لأن النموذج يحتوي على تباين منخفض في هذه المرحلة ويعمم البيانات جيدًا. يزيد تدريب النموذج الإضافي من تباين النموذج ويؤدي إلى الضبط الزائد.

الحذف العشوائي[2]

يشير "الحذف العشوائي" في الشبكات العصبية إلى عملية تجاهل عشوائي لعقد محددة في طبقة ما أثناء تدريب الشبكة. بمعنى آخر ، تتم إزالة الخلايا العصبية المختلفة مؤقتًا من الشبكة. أثناء التدريب ، يؤدي الحذف العشوائي إلى تغيير فكرة تعلم جميع أوزان الشبكة إلى تعلم جزء صغير فقط من أوزان الشبكة. من الشكل المقابل ، يمكن ملاحظة أنه في مرحلة التدريب القياسية ، يتم إشراك جميع الخلايا





[1] راجع الفصل الخامس من الكتاب.

[2] Dropout

العصبية ، ومن خلال الحذف العشوائي ، يتم مشاركة عدد قليل فقط من الخلايا العصبية المختارة في التدريب والباقي "متوقف". لذلك ، بعد كل تكرار ، يتم تنشيط مجموعات مختلفة من الخلايا العصبية لمنع بعض الخلايا العصبية من السيطرة على بعض الميزات. يساعدنا هذا النهج ، على الرغم من بساطته ، على تقليل التكرار وتمكين بنى شبكات أعمق وأكبر يمكنها عمل تنبؤات جيدة بشأن البيانات التي لم تراها الشبكة من قبل.

تسوية الجماعية

واحدة من المشاكل في تدريب الشبكة العصبية بالإضافة إلى الانحدار الاشتقاقي هي مشكلة تغيير المتغيرات الداخلية للشبكة. تنشأ هذه المشكلة لأن المعاملات تتغير باستمرار أثناء عملية التدريب ، والتي بدورها تغير قيم دوال التنشيط. يؤدي تغيير قيم الإدخال من الطبقات الأولية إلى الطبقات التالية إلى تقارب أبطأ أثناء عملية التدريب ، لأن بيانات التدريب للطبقات اللاحقة غير مستقرة. بمعنى آخر ، الشبكات العميقة هي مزيج من عدة طبقات ذات دوال مختلفة ، وكل طبقة لا تتعلم فقط كيفية تعلم التمثيل العام من بداية التدريب ، ولكن عليها أيضًا أن تتقن التغييرات المستمرة في توزيعات المدخلات وفقًا لما سبق. طبقات. بينما يقوم المُحسِّن بتحديث المعاملات على افتراض أنها لا تتغير في الطبقات الأخرى ويقوم بتحديث جميع الطبقات في نفس الوقت ، سيؤدي هذا إلى نتائج غير مرغوب فيها عند الجمع بين دوال مختلفة.

تم اقتراح **التسوية الجماعية**[1] للتغلب على هذه المشكلة لتقليل عدم الاستقرار وتحسين الشبكة. في هذه الطريقة ، نقوم بتسوية بيانات الإدخال للطبقة بحيث يكون لها متوسط صفر وانحراف معياري واحد. من خلال وضع التسوية الجماعية بين الطبقات المخفية وإنشاء خاصية تباين مشتركة ، نقوم بتقليل التغييرات الداخلية لطبقات الشبكة.

الشبكة العصبية المتكررة

الشبكات العصبية المتكررة (RNN[2]) هي نوع من الشبكات العصبية الاصطناعية المصممة لاكتشاف الأنماط في تسلسل البيانات ، مثل النص والجينوم والكتابة اليدوية والكلمات المنطوقة وبيانات السلاسل الزمنية وأسواق الأسهم وما إلى ذلك. الفكرة من وراء هذه الشبكات العصبية هي أنها تسمح للخلايا بالتعلم من الخلايا المرتبطة سابقًا. يمكن القول أن هذه الخلايا لها "ذاكرة" بطريقة ما. ومن ثم ، فإنهم يبنون معرفة أكثر تعقيداً من بيانات الإدخال.

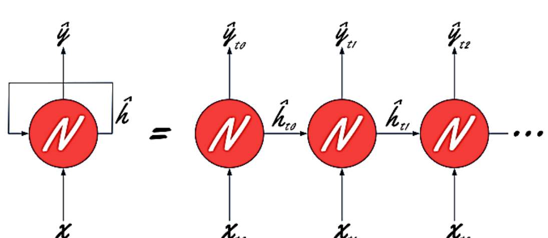
تقضي الشبكات العصبية المتكررة على أوجه القصور في الشبكات العصبية ذات التغذية الأمامية. وذلك لأن شبكات التغذية الامامية يمكنها فقط قبول مدخلات ذات حجم ثابت وتنتج

[1] Batch normalization

[2] Recurrent neural network

فقط مخرجات ذات حجم ثابت وغير قادرة على النظر في المدخلات السابقة بنفس الترتيب. من خلال النظر في المدخلات السابقة للتسلسل ، تكون الشبكة العصبية المتكررة قادرة على التقاط التبعيات عندما تكون شبكة العصبية امامية التغذية غير قادرة على ذلك.

تأخذ الشبكات العصبية المتكررة تسلسلاً كمدخلات وتقيم الشبكة العصبية لكل خطوة زمنية. يمكن اعتبار هذه الشبكات على أنها شبكة عصبية لها حلقة تسمح لها بالحفاظ على الحالة. عند التقييم ، تفتح الحلقة من خلال الخطوات الزمنية للتسلسل كما هو موضح في الشكل المقابل.



هذه الحلقات أو الروابط المتكررة هي سبب تسمية هذه الشبكات بالشبكات العصبية المتكررة. حقيقة أن شبكة المتكررة تتكون من حلقة تعني أنه يمكن إرجاع ناتج خلية عصبية واحدة في نقطة زمنية واحدة إلى نفس الخلية العصبية في نقطة زمنية أخرى. والنتيجة هي أن الشبكة لديها ذاكرة لعمليات التنشيط السابقة (وبالتالي المدخلات السابقة التي لعبت دورًا في هذا التنشيط).

يتضمن الإدخال لكل خطوة زمنية خصائص الخطوة الزمنية والحالة المخفية للشبكة العصبية المتكررة. يتم تحديث الحالة المخفية h في كل مرحلة زمنية ويتم نشرها إلى المرحلة التالية. بهذه الطريقة ، يمكن للشبكة العصبية المتكررة عرض المدخلات السابقة عبر الوضع المخفي.

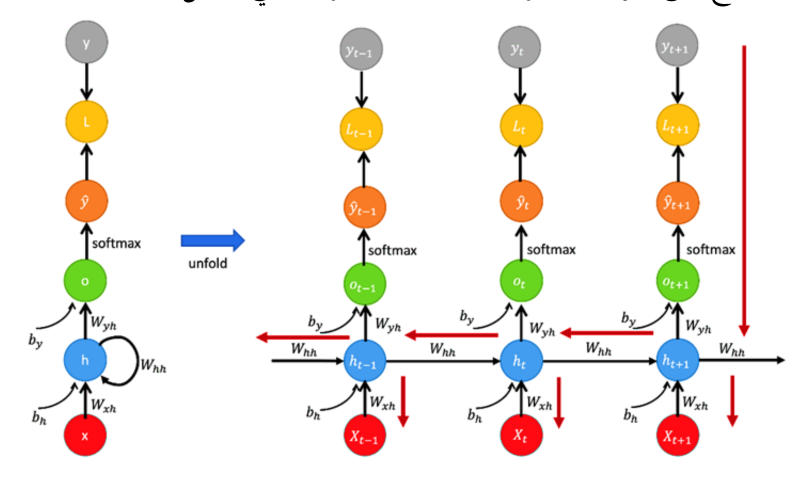
يتم تدريب الشبكات العصبية المتكررة باستخدام طريقة تسمى الانتشار الخلفي بمرور الوقت. في هذه الخوارزمية ، يجب حساب الانحدارات الاشتقاقية في كل خطوة زمنية باستخدام قاعدة السلسلة. عندما تكون التسلسلات طويلة ، يتطلب الإصدار اللاحق الكثير من الضرب. يمكن أن يؤدي هذا إلى مشكلة تسمى تلاشي أو انفجار التدرجات. عندما تتقلص التدرجات ، أو بعبارة أخرى ، يختفي التدرج ، ويتم تصغير تحديثات التعلم وتتوقف عن التعلم. في المقابل ، تؤدي التدرجات الكبيرة إلى قفزات في خطوات التعلم وتمنعها من الاقتراب من حل جيد. تحد مشكلة التلاشي والانفجار المتدرج من قدرة الشبكات العصبية المتكررة على تعلم التبعيات طويلة المدى. وبالتالي ، تم تطوير شبكات عصبية متكررة أخرى لمعالجة هذه المشكلة.

الانتشار الخلفي بمرور الوقت

يتم تدريب الشبكات العصبية المتكررة بمرور الوقت بواسطة نوع خاص من خوارزمية الانتشار الخلفي تسمى الانتشار الخلفي بمرور الوقت. مثل خوارزمية الانتشار الخلفي في شبكة العصبية امامية التغذية ، تستخدم هذه الخوارزمية قاعدة السلسلة لحساب التدرج. يعد الانتشار الخلفي في الشبكات العصبية المتكررة أكثر صعوبة بعض الشيء بسبب الطبيعة المتكررة للأوزان واختفائها بمرور الوقت. لأننا نحتاج إلى تمديد الرسم البياني الحسابي لـ RNN مرة واحدة

للحصول على التبعيات بين المتغيرات ومعاملات النموذج. بعد ذلك ، باستخدام الانتشار الخلفي وقاعدة السلسلة ، يتم حساب التدرجات وتخزينها. نظرًا لأن التسلسلات يمكن أن تكون طويلة ، فقد تكون التبعيات طويلة. على سبيل المثال ، بالنسبة لتسلسل من 1000 محرف ، يمكن أن يكون للمحرف الأول تأثير كبير على المحرف في الموقع النهائي. هذا ليس ممكنًا من الناحية الحسابية حقًا. لأنها تتطلب الكثير من الوقت والذاكرة.

في ما يلي ، سنصف الانتشار الخلفي بمرور الوقت رياضيًا. لفهم كيفية عمل هذا النهج ، يمكنك الاطلاع على نظرة عامة حول كيفية تدفق المعلومات في الشكل أدناه:



من أجل التمكن من استخدام الانتشار الخلفي بمرور الوقت في عملية تدريب الشبكة العصبية المتكررة ، يجب أولاً حساب دالة الخطأ:

$$L(\hat{y}, y) = \sum_{t=1}^{T} L_t(\hat{y}_t, y_t)$$

$$= -\sum_{t}^{T} y_t log\, \hat{y}_t$$

$$= -\sum_{t=1}^{T} y_t log\, [softmax(o_t)]$$

نظرًا لأن وزن w_{yh} مقسم في جميع المتتاليات الزمنية. ومن ثم ، يمكننا اشتقاقها في أي مرحلة وتجميعها معًا:

$$\frac{\partial L}{\partial w_{yh}} = \sum_{t}^{T} \frac{\partial L_t}{\partial w_{yh}}$$

$$= \sum_{t}^{T} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial o_t} \frac{\partial o_t}{\partial w_{yh}}$$

$$= \sum_{t}^{T} (\hat{y} - y_t) \otimes h_t$$

في هذه المعادلة $\frac{\partial o_t}{\partial w_{yh}} = h_t$ و$\otimes$الضرب الخارجي هما متجهان.

وبالمثل ، يمكننا الحصول على تدرج التحيز b_y:

$$\frac{\partial L}{\partial b_y} = \sum_{t}^{T} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial o_t} \frac{\partial o_t}{\partial b_y}$$

$$= \sum_{t}^{T} (\hat{y} - y_t)$$

بالإضافة إلى ذلك ، دعنا نستخدم L_{t+1} للإشارة إلى ناتج الخطوة الزمنية $t + 1$:

$$L_{t+1} = -y_{t+1} log \hat{y}_{t+1}$$

الآن ، سنراجع تفاصيل الانحدار الاشتقاقي w_{hh} فيما يتعلق بالوقت $t + 1$:

$$\frac{\partial L_{t+1}}{\partial w_{hh}} = \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial w_{hh}}$$

منذ الحالة المخفية h_{t+1} فيما يتعلق بالمعادلة المتكررة h_t:

$$h_t = \tanh\left(w_{xh}^T . x_t + w_{hh}^T . h_{t-1} + b_h\right)$$

يعتمد أيضا. لذلك ، في الخطوة الزمنية $t - 1 \rightarrow t$ ، يمكن الحصول على المشتق الجزئي فيما يتعلق بـ w_{hh} على النحو التالي:

$$\frac{\partial L_{t+1}}{\partial w_{hh}} = \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_t} \frac{\partial h_t}{\partial w_{hh}}$$

لذلك ، في الخطوة الزمنية $t + 1$ ، يمكننا حساب الانحدار الاشتقاقي واستخدام $t + 1$ إلى t من خلال إعادة الإرسال بمرور الوقت للحصول على الانحدار الكلي فيما يتعلق بـ w_{hh}

$$\frac{\partial L_{t+1}}{\partial w_{hh}} = \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k} \frac{\partial h_k}{\partial w_{hh}}$$

لاحظ أن $\frac{\partial h_{t+1}}{\partial h_k}$ هي نفسها قانون السلسلة. فمثلا:

$$\frac{\partial h_3}{\partial h_1} = \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1}$$

لاحظ أيضًا أنه نظرًا لأننا نعتبر مشتقة الدالة كمتجه ، فهي نتيجة مصفوفة (مصفوفة •Jacobin) تكون فيها جميع العناصر مشتقات جزئية. يمكننا إعادة كتابة التدرج أعلاه:

$$\frac{\partial L_{t+1}}{\partial w_{hh}} = \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \left(\prod_{j=k}^{t} \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial w_{hh}}$$

حيث:

$$\prod_{j=k}^{t} \frac{\partial h_{j+1}}{\partial h_j} = \frac{\partial h_{t+1}}{\partial h_k} = \frac{\partial h_{t+1}}{\partial h_t} \frac{\partial h_t}{\partial h_{t-1}} \cdots \frac{\partial h_{k+1}}{\partial h_k}$$

تتم إضافة التدرجات المتعلقة بـ w_{hh} معًا في جميع خطوات الانتشار الخلفي ، وأخيرًا يمكننا الحصول على الانحدارات الاشتقاقية التالية فيما يتعلق بـ w_{hh}:

$$\frac{\partial L}{\partial w_{hh}} = \sum_{t}^{T} \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k} \frac{\partial h_k}{\partial w_{hh}}$$

الآن دعنا نستخرج التدرج بالنسبة إلى w_{xh}. وبالمثل ، ضع في اعتبارك الخطوة الزمنية $t+1$ واحصل على الانحدار الاشتقاقي فيما يتعلق بـ w_{xh} على النحو التالي:

$$\frac{\partial L_{t+1}}{\partial w_{xh}} = \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial w_{xh}}$$

نظرًا لأن كل من h_t و x_{t+1} يساهمان في h_{t+1} ، فنحن بحاجة إلى h_t للانتشار الخلفي. إذا أخذنا في الاعتبار هذه الشراكة ، فسنحصل عليها:

$$\frac{\partial L_{t+1}}{\partial w_{xh}} = \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial w_{xh}} + \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_t} \frac{\partial h_t}{\partial w_{xh}}$$

لذلك ، من خلال جمع جميع المساهمات من $t+1$ إلى t عبر الانتشار الخلفي ، يمكننا الحصول على الانحدار الاشتقاقي في الوقت $t+1$:

• بالنظر إلى دالة التعيين $n-$ التالي للمتجه x إلى متجه الإخراج $m-$ التالي، $\boldsymbol{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$، تسمى مصفوفة جميع المشتقات الجزئية من الدرجة الأولى لهذه الدالة بمصفوفة Jacobin ($\boldsymbol{J}$):

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial L_{t+1}}{\partial w_{xh}} = \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k} \frac{\partial h_k}{\partial w_{xh}}$$

بالإضافة إلى ذلك ، يمكننا اعتبار المشتق بالنسبة إلى w_{xh} في المتتالية بأكملها:

$$\frac{\partial L}{\partial w_{xh}} = \sum_{t}^{T} \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k} \frac{\partial h_k}{\partial w_{xh}}$$

لا تنس أيضًا أن $\frac{\partial h_{t+1}}{\partial h_k}$ هي نفسها قانون السلسلة.

كما ذكرنا ، هناك مشاكل في تلاشي وانفجار الانحدار الاشتقاقي في الشبكة العصبية المتكررة العادية. بشكل عام ، هناك عاملان يؤثران على مقدار التدرجات: **الأوزان ودوال التنشيط** ، **أو بشكل أكثر دقة** ، **المشتقات** التي يمر منها الانحدار الاشتقاقي. في الشبكة العصبية المتكررة العادية ، ينتج تلاشي وانفجار الانحدار الاشتقاقي عن الاتصالات المتكررة. بتعبير أدق ، هاتان المشكلتان ترجعان إلى المشتق العائد $\frac{\partial h_{t+1}}{\partial h_k}$ الذي يحدث في المعادلة w_{xh} ويجب حسابه:

$$\prod_{j=k}^{t} \frac{\partial h_{j+1}}{\partial h_j} = \frac{\partial h_{t+1}}{\partial h_k} = \frac{\partial h_{t+1}}{\partial h_t} \frac{\partial h_t}{\partial h_{t-1}} \dots \frac{\partial h_{k+1}}{\partial h_k}$$

ويمثل ضرب المصفوفة على المتتالية.

نظرًا لأن الشبكة العصبية المتكررة العادية تحتاج إلى الحصول على الانحدار الاشتقاقي للانتشار الخلفي بتسلسل طويل (بقيم صغيرة مضروبة في المصفوفة) ، فإن مقدار التدرج ينخفض طبقة تلو الأخرى ويختفي في النهاية بعد بضع خطوات. لذلك ، لن تساعد المواقف البعيدة عن المرحلة الزمنية الحالية في حساب معاملات التدرج ، وهي نفس معاملات التعلم في الشبكة العصبية المتكررة.

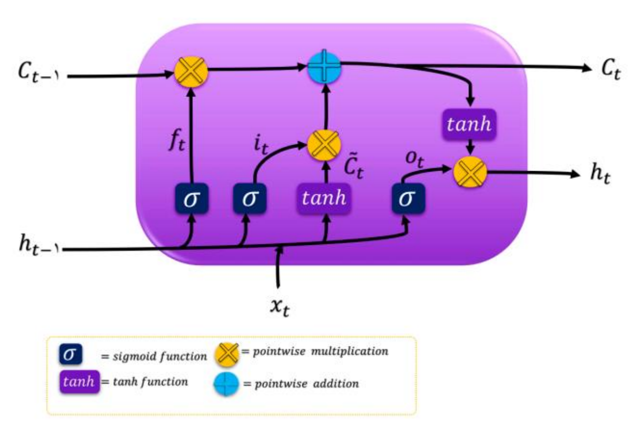
لا يقتصر تلاشي الانحدار على الشبكة العصبية المتكررة العادية. كما ذكرنا سابقًا ، تحدث أيضًا في الشبكات العصبية امامية التغذية. النقطة المهمة هي أن الشبكة العصبية المتكررة أكثر عرضة لهذه المشاكل لأنها عميقة جدًا. تظهر هاتان المشكلتان في النهاية أنه إذا تلاشى الانحدار الاشتقاقي ، فهذا يعني أن الحالات المخفية السابقة ليس لها تأثير حقيقي على الحالات المخفية التالية. بمعنى آخر ، لا يتم تعلم التبعية طويلة المدى. لحسن الحظ ، هناك عدة طرق لإصلاح مشكلة تلاشي الاشتقاق. يمكن أن يؤدي التهيئة المناسبة لمصفوفات الوزن إلى تقليل تأثير تلاشي الانحدار. يمكن أن يساعد التنظيم أيضًا. حل آخر مفضل أكثر من الحلين السابقين هو استخدام وظيفة التنشيط ReLU بدلاً من دوال التنشيط tanh او sigmoid. مشتق ReLU هو ثابت من 0 أو 1 ، لذلك على الأرجح لا يحتوي على مشكلة تلاشي الانحدار. الحل الأكثر شيوعًا والمستخدم على نطاق واسع اليوم هو استخدام شبكات الذاكرة قصيرة المدى طويلة المدى.

الذاكرة قصيرة المدى المستمرة لفترة أطول[1]

تم بناء شبكات الذاكرة قصيرة المدى المستمرة لفترة أطول (LSTMs) للتغلب على مشكلة تلاشي التدرج أو انفجاره. تحتوي LSTM على خلية حالة تتحكم ، بالإضافة إلى الحالة المخفية، في تدفق المعلومات بين المراحل الزمنية. يحتوي LSTM أيضًا على بوابات تُستخدم لتغيير الحالة وتشكيل مخرجات. يتم توفير نظرة عامة على خلية LSTM في الشكل 7-4.

تُستخدم بوابة النسيان لنسيان المعلومات غير ذات الصلة بحالة الخلية. بوابة النسيان للمصفوفة لها وزنها. مدخلات بوابة الإدخال هي الحالة المخفية السابقة h_{t-1} والمدخلات الحالية هي x_t. تُستخدم دالة sigmoid لتوليد ناتج بين صفر وواحد لكل عنصر في خلية الحالة. يتم تنفيذ الضرب الأولي بين خرج بوابة النسيان وخلية الحالة. تعني قيمة واحد عند إخراج بوابة النسيان الاحتفاظ الكامل بمعلومات العنصر في خلية الحالة ، بينما يعني الصفر النسيان الكامل للمعلومات الموجودة في عنصر خلية الحالة. معادلة بوابة النسيان كالتالي:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)$$



شكل 7ــ4. هيكل الخلية في LSTM

العملية الثانية في خلية LSTM هي بوابة الإدخال. تكتشف البوابة معلومات جديدة من مرحلة الوقت الحالية والحالة المخفية السابقة التي يجب تضمينها في حالة الخلية. يتم ذلك في جزأين: تحديد القيم المراد تحديثها ثم إنشاء القيم المراد تحديثها. يتم استخدام المتجه i_t أولاً لتحديد قيم المرشحين الجدد المحتملين لتضمينها في حالة الخلية. يحتوي المتجه المرشح C_t أيضًا على مصفوفة وزن خاصة به ويستخدم الحالة والمدخلات المخفية السابقة لتشكيل متجه بأبعاد مماثلة لخلية الحالة. لإنشاء هذا المتجه المرشح ، يتم استخدام دالة tanh كدالة غير خطية. يتم بعد ذلك إجراء عملية مضاعفة للعنصر بين متجه الإدخال i_t والمرشحين C_t لتحديد المعلومات

[1] Long Short-Term Memory Networks

الجديدة المراد تضمينها في خلية الحالة. أخيرًا ، يُضاف نتيجة الضرب إلى خلية الحالة. تظهر هذه العملية في المعادلات التالية:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = tanh(W_c.[h_{c-1}, x_t] + b_c)$$

تحدد بوابة النسيان والادخال كيفية تحديث خلية الحالة في كل خطوة زمنية. يتم تحديث خلية الحالة في خطوة زمنية من خلال المعادلة التالية:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

أخيرًا ، يجب تحديد ما هو موجود في الإخراج. الناتج النهائي لخلية LSTM هو الحالة المخفية h_t. أولاً ، تُستخدم sigmoid لحساب المتجه بقيم بين صفر وواحد لتحديد قيم خلية الحالة في الخطوة الزمنية. ثم نعطي قيمة خلية الحالة لطبقة tanh لمضاعفة قيمتها أخيرًا بإخراج طبقة sigmoid السابقة ، بحيث تتم مشاركة الأجزاء المرغوبة عند الإخراج. تظهر المعادلات التالية هذا الاتجاه:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

الطريقة التي يقلل بها LSTM من مشكلة تلاشي التدرج أو الانفجار هي في المعادلة:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

تمتلك LSTMs بُنية داخلية معقدة تتضمن عدة طبقات من الخلايا العصبية ويمكن اعتبارها شبكة واحدة. ومع ذلك ، يمكن أيضًا استخدامها كوحدات بناء لشبكة عصبية متكررة. يتم تحقيق ذلك عن طريق استبدال الطبقة المخفية في شبكة عصبية متكررة بوحدة LSTM. لقد حققت LSTMs نجاحًا كبيرًا في معالجة اللغة. على سبيل المثال ، هم الآن الشبكة القياسية المستخدمة لاكتشاف الكلام على الهواتف المحمولة.

الشبكة العصبية الالتفافية

ضع في اعتبارك شبكة تغذية عصبية متصلة بالكامل تأخذ صورة RGB بسيطة [256 × 256 × 3] كمدخل لها. وبناءً على ذلك ، فإن كل خلية عصبية بمفردها لها وزن 196608 = [256 × 256 × 3] وزن وهذا الوزن لخلية واحدة فقط !! من ناحية أخرى ، تتطلب البنى العميقة أعدادًا كبيرة من الخلايا العصبية والطبقات المخفية لتمثيل هياكل معقدة بدرجة كافية في بيانات الإدخال. هذا يعني أن طبيعة الاتصال الكامل لهذه الشبكات تستهلك قدرًا كبيرًا من الذاكرة ، خاصة للصور أو مقاطع الفيديو الكبيرة. بالإضافة إلى ذلك ، فإن عددًا كبيرًا من المعاملات يزيد

من ميل الشبكة إلى الضبط الزائد. لذلك ، لمعالجة هذه المشكلات ، تم تقديم الشبكات العصبية الالتفافية (CNN[1]) كتطوير شائع جداً للشبكات العصبية القياسية. الشبكات العصبية الالتفافية هي فئة من شبكات العصبية امامية التغذية التي تستخدم الطبقات الالتفافية لتحليل المدخلات باستخدام طبولوجيا الشبكة ، مثل الصور ومقاطع الفيديو. يعتمد اسم هذه الشبكات على الدالة الرياضية المسماة الالتفاف التي يستخدمونها في بُنيتهم. باختصار ، الشبكات الالتفافية هي شبكات عصبية تستخدم الالتفاف بدلاً من مضاعفة المصفوفة في طبقة واحدة على الأقل من طبقاتها.

هيكل الشبكة الالتفافية

في بنية الشبكة الالتفافية ، هناك جزءان رئيسيان:

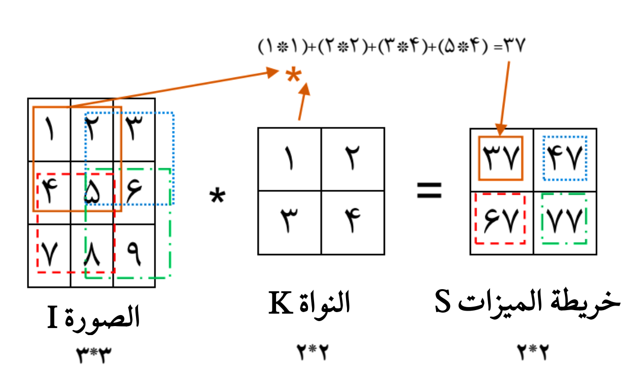
- **استخراج الميزة:** في هذا القسم ، يحدد الميزات المختلفة للصورة باستخدام الالتفاف والدمج.
- **التصنيف:** في هذا الجزء، باستخدام طبقة متصلة بالكامل ، يتم استخدام عملية استخراج الميزات ويتنبأ بفئة الصورة بناءً على الميزات المستخرجة في الخطوات السابقة. يمكن رؤية هيكلها العام في الشكل 7ـ5.

طبقة الالتفاف

بالنسبة للصورة ثنائية الأبعاد I ، يتم تعريف الالتفاف المنفصل على النحو التالي:

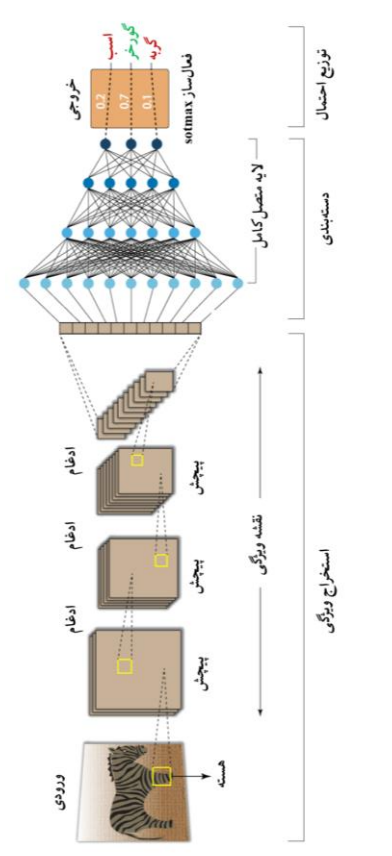
$$S(i,j) = (I * K)(i,j) \sum_{m} \sum_{n} I(i-m, j-n) K(m,n)$$

حيث $K(m,n)$ هي نواة ثنائية الأبعاد والمخرج $S(i,j)$ يسمى خريطة الميزات. بشكل حدسي، هذه العملية "تنزلق" الفلتر او الكيرنل (النواة) على طول الصورة I وتحسب مجموع الاوزان للكيرنل في كل موضع i و j من الصورة. يظهر مثال على مثل هذا الالتفاف ثنائي الأبعاد المنفصل في الشكل



[1] Convolutional neural network

المقابل. في الشبكة العصبية الالتفافية ، تصف النواة K الأوزان القابلة للتعلم لطبقة الالتفاف ، ويمكن أن تحتوي كل طبقة التفاف على العدد المطلوب من النوى ، وكل منها يؤدي إلى خريطة لميزات مخرجاتها.

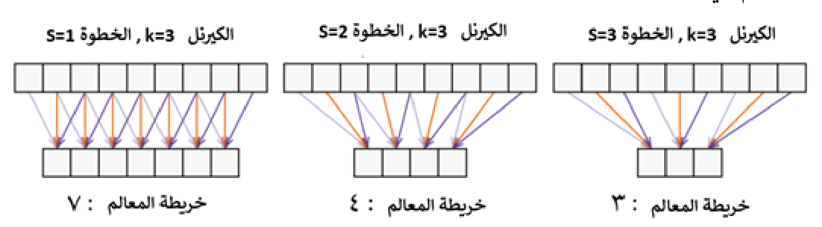


الشكل 7–5. نظرة عامة على هيكل الشبكة الالتفافية

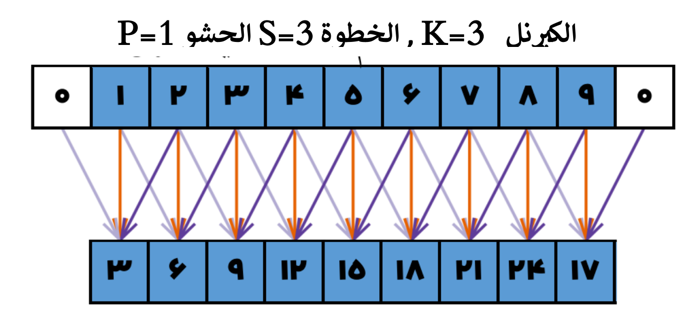
تحتوي كل طبقة من طبقات الالتفاف على مجموعة محددة من المعاملات الفائقة ، كل منها يحدد عدد الاتصالات وحجم الإخراج لخرائط الميزات:

- **حجم الفلتر:** يصف حجم الكيرنل K (يسمى أحياناً حجم الفلتر) حقل الاستلام الذي ينطبق على جميع مواقع الإدخال. تسمح زيادة هذه المعامل لطبقة الالتفاف بتلقي المزيد من المعلومات المكانية ، مع زيادة عدد أوزان الشبكة في نفس الوقت.
- **عدد الفلاتر:** يتوافق عدد النوى بشكل مباشر مع عدد المعاملات القابلة للتعلم وعمق D لحجم إخراج طبقة الالتفاف. مثلما ينتج كل نواة خريطة ميزات مخرجات منفصلة، فإن النوى D تنتج خريطة ميزات مخرجات بعمق D.
- **الخطوة:** كما أوضحنا سابقًا ، يمكن فهم الالتفاف على أنه ركيزة عن طريق "انزلاق" فلتر على حجم إدخال. ومع ذلك ، لا يجب أن يحدث "الانزلاق" على مسافة بكسل

واحد في كل مرة ، وهو ما تصفه الخطوة. تحدد الخطوة S عدد وحدات البكسل التي تنقلها النواة بين كل حساب لخاصية الإخراج. تنتج الخطوات الأكبر حجمًا خرائط ميزات مخرجات أصغر لأنه يتم إجراء عدد أقل من العمليات الحسابية. يظهر هذا المفهوم في الشكل أدناه:



- **الطبقات الصفرية**: نظرًا لتشغيل عملية الالتفاف ، تُستخدم الطبقات الصفرية للتحكم في الأبعاد بعد تطبيق فلاتر أكبر من 1 × 1 ولمنع فقدان المعلومات في الهوامش. بمعنى آخر ، غالبًا ما يتم استخدام الطبقة الصفرية للحفاظ على الأبعاد المكانية لطبقات الإدخال والإخراج كما هي. عن طريق إضافة مدخلات صفرية حول المحيط، يمكن تجنب انكماش الأبعاد المكانية عند الالتفاف. تعد قيمة الأصفار المضافة على كل جانب لكل بُعد مكاني معامل فائق إضافي P. يظهر مثال على الطبقات الصفرية في الشكل أدناه:



- **التمدد[1] (التوسع)**: التمدد d الذي تم إدخاله مؤخرًا هو معامل فائق أخر يسمح لطبقة الالتفاف أن يكون لها مجال استقبال أكثر كفاءة من الإدخال ، مع الحفاظ على حجم النواة ثابتًا. يتم الحصول على ذلك عن طريق إدخال المسافة d بين كل خلية من النواة. يستخدم الالتفاف القياسي ببساطة الدوران 0. ومن ثم فإن لها نواة مستمرة. من خلال زيادة المساحة ، يمكن لطبقة الالتفاف أن تشغل مساحة أكبر من الإدخال مع الحفاظ

[1] Dilation

على ثبات استهلاك الذاكرة. يظهر مفهوم الالتفافات الشعاعية[1] ، التي تسمى أحيانًا الالتفافات التصلبية ، في الشكل 6ـ7 بالتفافات مختلفة.

وفقًا لحجم الإدخال W ، حجم النواة او الفلتر K ، الخطوة S ، التمدد d والطبقة الصفرية P ، يتم حساب حجم الإخراج الناتج على النحو التالي:

$$W_o = \left\lfloor \frac{W + 2P - K - (K-1)(d-1)}{S} \right\rfloor + 1.$$

استخدام التفاف له ثلاث مزايا مهمة. أولاً، عادةً ما يكون للشبكات العصبية الالتفافية **اتصالات انفرادية**[2]. تستخدم شبكات العصبية امامية التغذية مصفوفة من المعاملات التي تصف العلاقة بين وحدات الإدخال والإخراج. هذا يعني أن كل وحدة إخراج متصلة بكل وحدة إدخال. ومع ذلك، فإن الشبكات العصبية الالتفافية لها اتصال انفرادي يتم الحصول عليه عن طريق تقليص النواة من المدخلات. على سبيل المثال، يمكن أن تحتوي الصورة على ملايين أو آلاف وحدات البكسل، ولكن أثناء معالجتها باستخدام النواة، يمكننا تحديد المعلومات المفيدة التي تتكون من عشرات أو مئات من وحدات البكسل. هذا يعني أنه يتعين علينا تخزين عدد أقل من المعاملات التي لا تقلل من الحاجة إلى الذاكرة فحسب، بل تعمل أيضًا على تحسين الأداء الإحصائي للنموذج. ثانيًا، تستخدم الشبكات العصبية الالتفافية **مشاركة المعاملات**[3]. أي أنهم يعيدون استخدام نفس المعاملات للعديد من الدوال. تجلب المعاملات المشتركة أيضًا الميزة الرئيسية الأخيرة، وهي **التقارب**. التقارب[4] يعني أنهفي حالة تهجير المدخلات، يتم إزاحة المخرجات بنفس الطريقة. هذه الميزة ضرورية لمعالجة البيانات ثنائية الأبعاد، لأنه إذا تم نقل صورة أو جزء منها إلى موقع آخرفي الصورة، فسيكون لها نفس العرض.

طبقة الدمج

من أجل الحفاظ على قيمة المعاملات منخفضة وزيادة مجال القبول الفعال للمخرجات وفقًا للإدخال ، يمكن أن يكون استخدام شكل خاص من أخذ العينات المكانية ، يسمى **الدمج** او **التجميع** ، بعد عدة طبقات من الالتفاف في الشبكة مفيدًا. يمكن فهم الدمج ، على غرار الالتفاف، بشكل حدسي على أنه آلية كيرنل زلقة ، ذات معايير مماثلة ، مثل درجة الميل وحجم النواة. الاختلاف الرئيسي هو أنه يحسب تكامل دالة ثابتة في مدخلاته ، والتي عادة ما تكون **الحد الأقصى** للعملية. يتضمن الشكل الأكثر شيوعًا للتكامل نواة [2 × 2] بالخطوة 2. عندما يتم

---

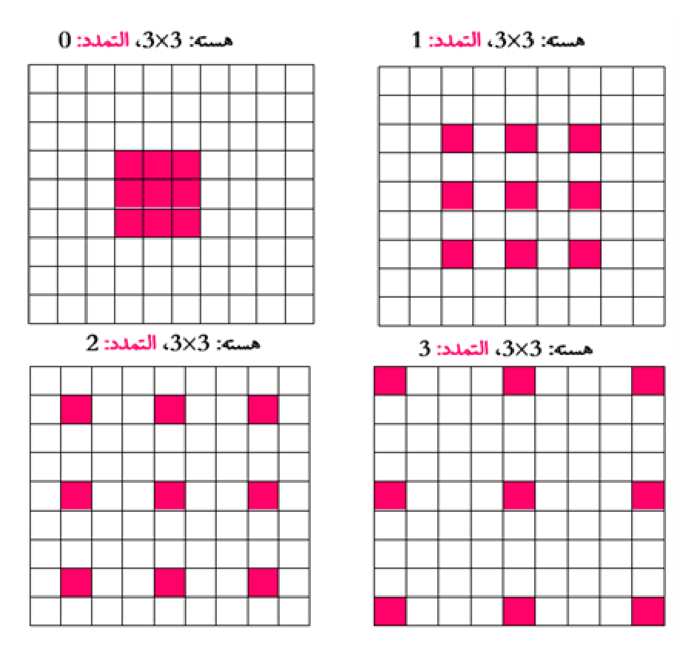
[1] atrous convolutions

[2] Sparse interactions

[3] Parameter sharing

[4] Equivariance

تطبيق هذه النواة على حجم الإدخال باستخدام دالة الحد الأقصى ، تتم معالجة التصحيحات غير المتداخلة [2. 2] لحجم الإدخال بشكل فعال ، مع الاحتفاظ فقط بأكبر قيمة في خريطة سمة الإخراج وتجاهل 75٪ من بيانات الإدخال. يمكن إجراء الانتشار الخلفي لتحقيق أقصى قدر من الدمج ببساطة عن طريق توجيه الانحدار الاشتقاقي فقط من خلال الإدخال بأعلى قيمة في التمريرة الخلفية.



شكل 7ـ6. فراخش روی ورودی دو بعدی با اندازه‌های مختلف.

نظرًا لأن هذه الدالة ثابتة ، فإنها لا تتطلب أي معاملات قابلة للتعليم وبالتالي لا تزيد من استهلاك الذاكرة وقدرة نموذج بُنية CNN مقارنةً **بالتفافات الخطوة**[1]. ومع ذلك ، يبدو أن هياكل CNN الحديثة تتجنب استخدام الدمج للاختزال[2] ، وبدلاً من ذلك تقترح دائمًا استخدام الالتفافات في الخطوات لتقليل الأبعاد المكانية. يبدو أن هذا مهم بشكل خاص عند تدريب النماذج التوليدية مثل **شبكات الخصومة التوليدية**[3].

الطبقات غير الخطية

كما ذكرنا سابقًا ، يمكننا استخدام الشبكات العصبية العميقة لاكتشاف أنواع مختلفة من الصور. ومع ذلك ، إذا استخدمنا الطبقات الخطية فقط ، مثل الالتفاف ، فيمكننا العثور على تحويل

---

[1] strided convolutions

[2] downsampling

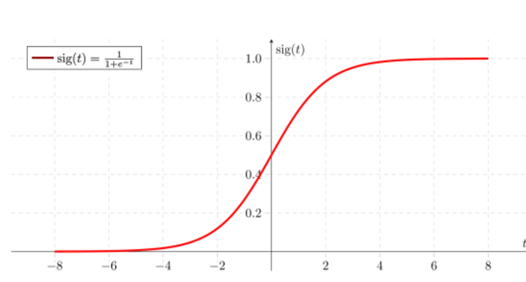
[3] Generative Adversarial Networks (GANs)

خطي من طبقة واحدة لاستبدال الشبكة العصبية العميقة. بمعنى آخر ، لا يهم أن تتكون الشبكة من عدة عمليات خطية ، فالنظام بأكمله ليس أقوى من الانحدار الخطي البسيط. على هذا النحو، لا توجد طريقة للاستفادة من الشبكات العصبية العميقة. لذلك ، نظرًا لأن الالتواء عملية خطية وأن الصور غير خطية ، فغالبًا ما يتم وضع الطبقات غير الخطية مباشرة بعد طبقة الالتفاف لإنشاء علاقة غير خطية بين المدخلات والمخرجات.

في الشبكات العصبية ، يتم إدخال اللاخطية باستخدام مفهوم دالة التنشيط. هناك عدة أنواع من الدوال غير الخطية ، وأشهرها:

- **سيكمويد:** يتم تعريف دالة التنشيط سيكمويد على النحو التالي:

$\sigma(x) = \frac{1}{1+e^{-x}}$, الذي يعيّن الإدخال بالقيمة الحقيقية لـ x في النطاق بين 0 و 1. تظهر دالة التنشيط سيكمويد في الشكل المقابل.



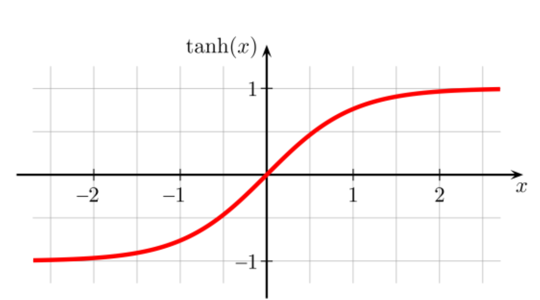
في الشبكات العصبية المبكرة، كانت سيكمويد خيارًا شائعًا. ومع ذلك ، فإن دالة التنشيط سيكمويد لها عيوب كبيرة. العيب الرئيسي لـسيكمويد هو أنه يشبع وبالتالي يوفر فقط تدرجات قريبة جداً من الصفر في هذه المناطق ، مما يمنع بشكل فعال التدرج من الانتقال عبر هذه الخلايا العصبية إلى جميع المدخلات بواسطة خوارزمية الانتشار الخلفي. بالإضافة إلى ذلك، الإخراج لدالة تنشيط سيكمويد ليست صفرية المركز[1]، والتي يمكن أن تؤدي إلى ديناميكيات غير مرغوب فيها أثناء الانحدار الاشتقاقي. لذلك ، لا يُسمح دائمًا باستخدام دالة التنشيط سيكمويد للخلايا العصبية المخفية. ومع ذلك ، بالنسبة للخلايا العصبية الناتجة ، يمكن أن يكون النطاق بين 0 و 1 مفيداً. على سبيل المثال ، لتفسير التنبؤات على أنها احتمالات.

- **الدالة الزائدية (tanh):** ترتبط دالة تنشيط تانتش ارتباطًا وثيقًا بدالة التنشيط سيكمويد وشكلها الرياضي هو كما يلي:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1.$$

[1] zero-centered

كما يمكن رؤيته في المعادلة أعلاه ، فإن tanh هو ببساطة نسخة مصغرة من sigmoid. ومع ذلك ، فهو صفري المركز. لذلك ، فإنه لا يظهر بعض المشاكل التي يعاني منها سيكمويد. يظهر المنشط تانتش في الشكل المقابل.

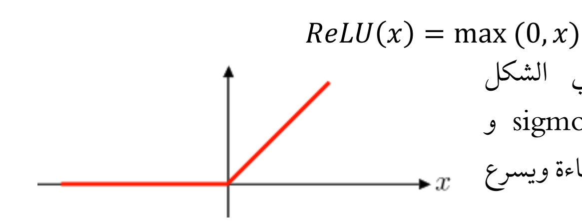


- **وحدة المعدل الخطي (ReLU):**

يتم تعريف دالة التنشيط ReLU على النحو التالي:

$$ReLU(x) = \max(0, x)$$

والتي يمكن رؤيتها في الشكل المقابل. بالمقارنة مع sigmoid و tanh ، فإن ReLU أكثر كفاءة ويسرع التقارب.



خلاصة الفصل السابع

- يصف التعلم العميق الخوارزميات التي تحلل البيانات بهيكل منطقي ، على غرار الطريقة التي يستنتجها الإنسان.
- يعتمد جوهر التعلم العميق على طريقة تكرارية لتدريب الآلات لتقليد الذكاء البشري.
- في التعلم العميق ، لا نحتاج إلى برمجة صريحة لكل شيء.
- الخلايا العصبية الاصطناعية هي اللبنات الرئيسية للشبكات العصبية الاصطناعية.
- تجمع الشبكات العصبية أمامية التغذية عدة خلايا عصبية لتشكيل رسم بياني اتجاهي بدون دائرة.
- الغرض من تحسين الشبكات العصبية امامية التغذية هو العثور تلقائيًا على الأوزان والتحيزات التي تقترب من إخراج الشبكة المستهدفة y مع الإدخال x.
- ربما تكون خوارزمية الانتشار الخلفي هي اللبنة الأساسية في الشبكة العصبية.
- يعد الانتشار الخلفي في الأساس طريقة ذكية لحساب التدرجات بشكل فعال في الشبكات العصبية متعددة الطبقات.
- إحدى العوائق الرئيسية لتحسين الشبكات العصبية من خلال الانتشار الخلفي وتقليل التدرجات هي مشكلة تلاشي الانحدار.
- الغرض من الانحدار الاشتقاقي العشوائي هو تسريع عملية التعلم مع تغيير طفيف في إجراء الانحدار الاشتقاقي القياسي.

- الهدف من مُحسِّن معدل التعلم التكيفي هو حل مشكلة إيجاد معدل التعلم الصحيح.
- في الشبكات العصبية ، يجب اختيار القيم الأولية للأوزان بعناية فائقة.
- تُستخدم مجموعات التحقق من الصحة في الشبكات العصبية بشكل شائع لضبط المعاملات الفائقة للنموذج مثل بُنية الشبكة أو معدلات التعلم.
- يشير "الحذف العشوائي" في الشبكات العصبية إلى عملية تجاهل عُقد معينة عشوائية في طبقة أثناء تدريب الشبكة.
- الشبكات العصبية المتكررة هي نوع من الشبكات العصبية الاصطناعية المصممة لاكتشاف الأنماط في تسلسل البيانات.
- تم بناء شبكات ذاكرة طويلة المدى للتغلب على مشكلة تلاشي التدرجات أو انفجارها.
- الشبكات العصبية الالتفافية هي فئة من شبكات العصبية امامية التغذية التي تستخدم الطبقات الالتفافية لتحليل المدخلات باستخدام طبولوجيا الشبكة ، مثل الصور ومقاطع الفيديو.

مصادر إضافية لمزيد من القراءة

Buduma, N., & Locascio, N. (2017). Fundamentals of deep learning: Designing next-generation machine intelligence algorithms. " O'Reilly Media, Inc.".

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Gulli, A., & Pal, S. (2017). Deep learning with Keras. Packt Publishing Ltd.

Osinga, D. (2018). Deep learning cookbook: practical recipes to get started quickly. " O'Reilly Media, Inc.".

Ramsundar, B., & Zadeh, R. B. (2018). TensorFlow for deep learning: from linear regression to reinforcement learning. " O'Reilly Media, Inc.".

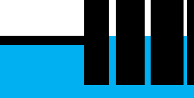
Trask, A. W. (2019). Grokking deep learning. Simon and Schuster.

Vazan, Milad. (2021). Deep learning: principles, concepts and approaches.

Wani, M. A., Bhat, F. A., Afzal, S., & Khan, A. I. (2020). Advances in deep learning. Springer.

8

التعلم غير الخاضع للاشراف



الاهداف:

- ما هو التعلم غير الاشرافى وما فوائده؟
- التعرف على التجميع وأنواع مختلفة من الخوارزميات؟
- الفرق بين اختيار الميزة واستخراج الميزة.
- تقليل الأبعاد الخطية وغير الخطية.
- التعرف على autoencoder.
- الفرق بين نموذج الإنتاج والفاصل.

التعلم غير الخاضع للإشراف وضعف التعلم الخاضع الإشراف

التعلم الخاضع للإشراف فعال للغاية في تحسين أداء المهام باستخدام مجموعات البيانات ذات العلامات المتعددة. على سبيل المثال ، ضع في اعتبارك مجموعة بيانات كبيرة جدًا من صور الكائنات التي تم وضع علامة عليها كل صورة. إذا كانت مجموعة البيانات كبيرة بما يكفي ، إذا قمنا بتدريبها جيدًا بما يكفي باستخدام خوارزميات التعلم الآلي الصحيحة (الشبكات العصبية الالتفافية) وباستخدام جهاز كمبيوتر قوي ، يمكننا بناء نموذج تصنيف للصور قائم على التعلم يخضع للإشراف جيدًا. نظرًا لأنه يتم تدريب الخوارزمية الخاضعة للاشراف ، يمكنها قياس أدائها (عبر دالة التكلفة) من خلال مقارنة علامة الصورة المتوقعة مع علامة الصورة الفعلية الموجودة لدينا في مجموعة البيانات. تحاول الخوارزمية صراحة تقليل دالة التكلفة هذه ؛ بحيث يكون الخطأ في الصور التي لم يتم رؤيتها من قبل (مجموعة الاختبار) أقل ما يمكن. هذا هو السبب في أن العلامات قوية جدًا ، فهي تساعد في توجيه الخوارزمية من خلال توفير مقياس للخطأ. تستخدم الخوارزمية مقياس الخطأ لتحسين أدائها بمرور الوقت. بدون هذه العلامات ، لا تعرف الخوارزمية مدى نجاحها في تصنيف الصور بشكل صحيح. ومع ذلك ، في بعض الأحيان تكون تكلفة وضع العلامات يدويًا على مجموعة بيانات عالية جدًا.

بالإضافة إلى ذلك ، على الرغم من قوة نماذج التعلم الخاضع للإشراف ، إلا أنها محدودة في تعميم المعرفة خارج نطاق المعرفة التي تم تدريبهم عليها. نظرًا لأن معظم بيانات العالم غير مسماة ، فإن قدرة الذكاء الاصطناعي على توسيع وظائفه لتشمل أمثلة غير مرئية من قبل محدودة باستخدام التعلم الخاضع للإشراف. بمعنى آخر ، يعد التعلم الخاضع للإشراف أمرًا رائعًا لحل مشاكل **الذكاء الاصطناعي المحدودة**[1]**(الضعيفة)** ، ولكنه ليس رائعًا لحل **مشكلات الذكاء الاصطناعي القوية**.

بعبارة أخرى ، التعلم غير الخاضع للإشراف يضيء حقًا للمشكلات التي تكون فيها الأنماط غير معروفة أو تتغير باستمرار أو ليس لدينا مجموعات بيانات ذات علامات كافية لها. يعمل التعلم غير الخاضع للإشراف ، بدلاً من الاسترشاد بالعلامات ، من خلال تعلم البنية الأساسية للبيانات التي يتم التدريب عليها. يقوم التعلم غير الإشرافي بذلك من خلال محاولة تمثيل البيانات التي يتم التدريب عليها بمجموعة من المعاملات. من خلال القيام بهذا **التعلم التمثيلي**[2]، يمكن للتعلم غير الخاضع للإشراف تحديد أنماط مميزة في مجموعة البيانات. في مثال مجموعة بيانات الصورة (هذه المرة غير مسماة) ، قد يكون التعلم غير الخاضع للإشراف قادرًا على تحديد

---

[1] Narrow AI

[2] representation learning

الصور وتجميعها بناءً على مدى تشابهها مع بعضها البعض وكيف تختلف عن الآخرين. على سبيل المثال ، يتم تجميع كل الصور التي تشبه الكراسي معًا ، ويتم تجميع كل الصور التي تشبه القطط معًا. بالطبع ، لا يستطيع التعلم غير الإشرافي نفسه تسمية هذه المجموعات بـ "الكراسي" أو "القطط". ومع ذلك ، الآن بعد أن تم تجميع الصور المتشابهة معًا ، أصبح لدى البشر مهمة أبسط بكثير تتمثل في وضع العلامات. بدلاً من وضع علامات على ملايين الصور يدويًا ، يمكن للبشر تحديد جميع المجموعات المنفصلة يدويًا وتطبيق هذه العلامات على جميع أعضاء كل مجموعة.

وبالتالي ، فإن التعلم غير الإشرافي يجعل المشكلات التي كانت غير قابلة للحل في السابق أكثر قابلية للحل وأكثر مرونة في العثور على الأنماط المخفية ، سواء في البيانات السابقة المتاحة للتدريب أو في البيانات المستقبلية. حتى لو كان التعلم غير الإشرافي أقل مهارة في حل مشكلات معينة (مشاكل محدودة للذكاء الاصطناعي) من التعلم الخاضع للإشراف ، فمن الأفضل التعامل مع المشكلات المفتوحة أكثر من الذكاء الاصطناعي القوي وتعميم هذه المعرفة. *والأهم من ذلك، أن التعلم غير الخاضع للإشراف يمكن أن يحل العديد من المشكلات الشائعة التي يواجهها علماء البيانات عند تطوير حلول التعلم الآلي.*

الذكاء الاصطناعي الضعيف والقوي

اليوم ، الذكاء الاصطناعي على شفاه الجميع ، ولا يمر يوم لم نسمع فيه عن الذكاء الاصطناعي. ومع ذلك ، فإن الحديث عن الذكاء الاصطناعي غالبًا ما يؤدي إلى سوء الفهم. هذا لأنه لا يوجد تعريف واضح للذكاء الاصطناعي. قد يجعل المساعدون الشخصيون مثل Siri أو Amazon Alexa أو Google Home بعض المستخدمين يعتقدون أنهم يتحدثون معهم أو يفهمونهم.

بشكل عام ، يتكون الذكاء الاصطناعي من خوارزميات متقدمة تتبع دالة رياضية يمكنها إجراء عمليات معقدة شبيهة بالإنسان. تشمل الأمثلة الفهم البصري والتعرف على الكلام واتخاذ القرار والترجمة بين اللغات. بشكل عام ، هناك قضيتان فكريتان في الذكاء الاصطناعي: الذكاء الاصطناعي الضعيف والذكاء الاصطناعي القوي. يُعتبر المساعدون الشخصيون المزعومون الأكثر اهتمامًا اليوم ، Siri أو Amazon Alexa أو Google Home ، برامج ذكاء اصطناعي ضعيفة لأنهم يعملون في مجموعة محدودة من الوظائف المحددة مسبقًا. حتى برامج الشطرنج المتقدمة تعتبر ذكاءً اصطناعيًا ضعيفًا. يبدو أن هذا التصنيف متجذر في الاختلاف بين البرمجة الخاضعة للإشراف وغير الخاضعة للإشراف. غالبًا ما يكون للمساعدين الشخصيين والشطرنج استجابة مخططة. إنهم يصنفون بناءً على أشياء مشابهة لما يعرفونه (من خلال التعلم من البيانات). إنها تقدم تجربة شبيهة بتجربة الإنسان ، لكنها مجرد محاكاة. إذا طلبت من Alexa تشغيل التلفزيون ، فسيفهم المبرمج الكلمات الرئيسية مثل on and TV. تستجيب الخوارزمية

من خلال تشغيل التلفزيون ، ولكنها تستجيب فقط لبرامجها. بمعنى آخر ، لا أحد يفهم معنى ما تقوله.

من ناحية أخرى ، هناك آلات لها عقولها الخاصة التي يمكنها اتخاذ قرارات مستقلة دون تدخل بشري. يمكن اعتبار هذه البرامج ذكاء اصطناعيًا قويًا. بمعنى آخر ، يشير الذكاء الاصطناعي القوي إلى الآلات أو البرامج التي لها عقولها الخاصة ويمكنها أداء المهام المعقدة بمفردها دون تدخل بشري. يتمتع الذكاء الاصطناعي القوي بخوارزمية معقدة تساعد الأنظمة على الأداء الجيد في مجموعة متنوعة من المواقف ، ويمكن للآلات ذات الذكاء الاصطناعي القوي اتخاذ قرارات مستقلة دون تفاعل بشري. يمكن للآلات القوية ذات الذكاء الاصطناعي أداء المهام المعقدة بمفردها ، تمامًا مثل البشر.

يعمل الذكاء الاصطناعي القوي الذي نشاهده في العديد من الأفلام مثل الدماغ. لا يصنف ، ولكنه يستخدم التجميع والاتصال لمعالجة البيانات. باختصار ، هذا يعني أنه لا توجد إجابة محددة لكلماتك الرئيسية. تحاكي الدالة النتيجة ، لكن في هذه الحالة ، لسنا متأكدين من النتيجة. مثل التحدث إلى إنسان ، يمكنك تخمين ما يجيب به الشخص ، لكنك لا تعرف على وجه اليقين. على سبيل المثال ، قد يسمع الجهاز عبارة "صباح الخير" ويبدأ في توصيله مع تشغيل آلة صنع القهوة. إذا كان الكمبيوتر لديه هذه القدرة ، فيمكنه نظريًا سماع "صباح الخير" ويقرر تشغيل آلة صنع القهوة.

الفرق بين الذكاء الاصطناعى القوى والضعيف

المعنى

الذكاء الاصطناعي القوي هو الشكل النظري للذكاء الاصطناعي ويقوم على افتراض أن الآلات يمكنها بالفعل تطوير الذكاء البشري بنفس الطريقة التي يتطور بها الإنسان. يشير الذكاء الاصطناعي القوي إلى آلة افتراضية توضح القدرات الإدراكية للإنسان. من ناحية أخرى ، يعد الذكاء الاصطناعي الضعيف نوعًا من الذكاء الاصطناعي يشير إلى استخدام الخوارزميات المتقدمة لأداء مهام محددة لحل المشكلات أو التفكير والتي لا تغطي النطاق الكامل للقدرات المعرفية البشرية.

الوظيفة

الأداء في الذكاء الاصطناعي الضعيف محدود مقارنةً بالذكاء الاصطناعي القوي. لا يحقق الذكاء الاصطناعي الضعيف الوعي الذاتي أو لا يُظهر نطاق القدرات الإدراكية البشرية التي قد يمتلكها الإنسان. يشير الذكاء الاصطناعي الضعيف إلى الأنظمة التي تمت برمجتها لأداء مجموعة واسعة من المهام ولكنها تعمل ضمن وظائف محددة مسبقًا أو محددة مسبقًا. من ناحية أخرى ، يشير

الذكاء الاصطناعي القوي إلى الآلات التي تظهر ذكاء الإنسان. الفكرة هي تطوير الذكاء الاصطناعي إلى الحد الذي يتفاعل فيه البشر مع الآلات التي يتم دفعها بوعي وذكاء ومع العواطف والوعي الذاتي.

الهدف

الهدف من الذكاء الاصطناعي الضعيف هو إنشاء تقنية تسمح للآلات وأجهزة الكمبيوتر بأداء مهام محددة لحل المشكلات أو التفكير بشكل أسرع بكثير من الإنسان. الغرض من الذكاء الاصطناعي القوي هو تطوير الذكاء الاصطناعي إلى الحد الذي يمكن اعتباره ذكاءً بشريًا حقيقيًا. الذكاء الاصطناعي القوي هو نوع لم يوجد بعد في شكله الحقيقي.

ملخص للذكاء الاصطناعي القوي مقابل الذكاء الاصطناعي الضعيف

باختصار ، الذكاء الاصطناعي القوي هو في الأساس نوع من الذكاء الاصطناعي المتقدم بما يكفي لاعتباره ذكاءً حقيقيًا. يعتمد الذكاء الاصطناعي القوي على فرضية أن آلة الحوسبة جيدة التنظيم لديها عقل يفهم ويفكر وله غرض مثل العقل البشري. من ناحية أخرى ، لا يحقق ضعف الذكاء الاصطناعي الوعي الذاتي أو يظهر مجموعة واسعة من القدرات الإدراكية البشرية. ومن ثم ، لا يمكن اعتبار برامج الذكاء الاصطناعي الضعيفة ذكية ، لأنها في الحقيقة لا تستطيع التفكير واتخاذ القرار بمفردها مثل البشر.

التعلم غير الإشرافي وتحسين استراتيجيات التعلم الآلي

ترجع النجاحات الأخيرة في التعلم الآلي إلى توفر كميات كبيرة من البيانات ، والتقدم في أجهزة الكمبيوتر ، والتقدم في خوارزميات التعلم الآلي. لكن هذه النجاحات اقتصرت على قضايا الذكاء الاصطناعي مثل تصنيف الصور ورؤية الكمبيوتر والتعرف على الكلام ومعالجة اللغة الطبيعية والترجمة الآلية.

لحل مشاكل الذكاء الاصطناعي الطموحة ، يجب أن نثبت قيمة التعلم غير الخاضع للإشراف. دعونا نلقي نظرة على بعض التحديات الأكثر شيوعًا التي يواجهها العلماء عند بناء الحلول وكيف يمكن أن يساعدهم التعلم غير الخاضع للإشراف.

البيانات الموسومة غير كافية

"أعتقد أن الذكاء الاصطناعي يشبه بناء مركبة فضائية. أنت بحاجة إلى محرك كبير وكثير من الوقود. إذا كان لديك محرك كبير وكمية صغيرة من الوقود ، فلا يمكنك الدخول إلى المدار. إذا كان لديك محرك صغير و طن من الوقود ، لا يمكنك حتى رفعه "تحتاج إلى محرك كبير وكثير من الوقود لبناء صاروخ." – أندرو آنج[1]

---

[1] Andrew Ng

إذا كان التعلم الآلي عبارة عن مركبة فضائية ، فستكون البيانات وقودها. بدون الكثير من البيانات، لا يمكن للمركبة الفضائية الطيران. لذلك ، لاستخدام خوارزميات التعلم الخاضع للإشراف ، نحتاج إلى الكثير من البيانات المصنفة التي يصعب إنتاجها ومكلفتها.

باستخدام التعلم غير الإشرافي ، يمكننا تلقائيًا وضع علامة على العينات غير المسماة. الطريقة التي تعمل بها هي أننا **نجمع**[1] جميع العينات ثم نطبق العلامات من العينات المصنفة على العينات غير المسماة في نفس المجموعة. تتلقى العينات غير المصنفة تسمية العينات الأكثر تشابهًا.

لعنه الأبعاد

في مساحة كبيرة جدًا ، تحتاج خوارزميات التعلم الخاضعة للإشراف إلى تعلم كيفية فصل النقاط من أجل إنشاء تقريب دالي لاتخاذ قرارات جيدة. ومع ذلك ، عندما تكون الميزات كثيرة ، يصبح البحث مكلفًا للغاية ، من الناحية الزمنية والحسابية. في بعض الحالات ، من المستحيل إيجاد حل جيد بالسرعة الكافية. تُعرف هذه المشكلة بالصعوبة متعددة الأبعاد (**لعنة الابعاد**) التي يكون التعلم غير الإشرافي مناسبًا للمساعدة في إدارتها. من خلال تقليل الأبعاد ، يمكننا العثور على أبرز الميزات في مجموعة الميزات الرئيسية ، وتقليل عدد الأبعاد إلى رقم أكثر قابلية للإدارة مع فقدان القليل جدًا من المعلومات المهمة في العملية ، ثم نستخدم خوارزميات خاضعة للإشراف لأداء فعال لتقريب الأداء الجيد.

هندسة الميزات

تعد هندسة الميزات إحدى المهام الرئيسية التي يؤديها علماء البيانات. بدون الميزات الصحيحة، لن تتمكن خوارزمية التعلم الآلي من عزل نقاط كافية في الفضاء لاتخاذ قرارات جيدة بشأن العينات غير المرئية. ومع ذلك ، فإن هندسة الميزات عادة ما تكون متطلبة للغاية. لأنه يتطلب من البشر تصميم النوع الصحيح من الميزات بشكل إبداعي. بدلاً من ذلك ، يمكننا استخدام التعلم التمثيلي من خوارزميات التعلم غير الإشرافية للتعرف تلقائيًا على الأنواع المناسبة من تمثيل السمات للمساعدة في حل المشكلة المطروحة.

لإنشاء تمثيلات لميزات جديدة ، يمكننا استخدام شبكة عصبية غير متكررة لأداء التعلم التمثيلي ؛ حيث يتوافق عدد الخلايا العصبية في طبقة الإخراج مع عدد الخلايا العصبية في طبقة الإدخال. تُعرف هذه الشبكة العصبية **بالتشفير الذاتي**[2] وتعيد إنتاج الميزات الرئيسية بفعالية وتتعلم تمثيلًا جديدًا باستخدام الطبقات المخفية بينها. تتعرف كل طبقة مخفية من المشفر الذاتي على تمثيل الميزات الرئيسية ، ويتم بناء الطبقات اللاحقة على التمثيل الذي تعلمته الطبقات

[1] clustering

[2] autoencoder

السابقة. طبقة تلو طبقة ، يتعلم المشفر التلقائي تمثيلات أكثر تعقيدًا من أمثلة أبسط. طبقة المخرجات هي آخر تمثيل تم تعلمه حديثًا للميزات الرئيسية. أخيرًا ، يمكن استخدام هذا التمثيل المتعلم كمدخل لنموذج التعلم الخاضع للإشراف لتحسين خطأ التعميم..

التعلم التمثيلي

يعتمد أداء أي نموذج للتعلم الآلي بشكل كبير على التمثيلات التي يتعلمها لتوليد المخرجات. هذا التمثيل المتعلم ، بدوره ، يعتمد بشكل مباشر على النموذج وما يتم تغذيته كمدخلات. تخيل مهندسًا يصمم نموذجًا للتعلم الآلي للتنبؤ بالخلايا الخبيثة بناءً على فحوصات الدماغ. لتصميم النموذج ، يجب أن يعتمد المهندس بشكل كبير على بيانات المريض ، لأن جميع الإجابات موجودة هنا. تصف كل ملاحظة أو ميزة في تلك البيانات خصائص المريض. يجب أن يتعلم نموذج التعلم الآلي الذي يتنبأ بالنتائج كيف ترتبط كل سمة بنتائج مختلفة: حميدة أو خبيثة. لذلك إذا كان هناك أي تشويش أو تناقض في البيانات ، فقد تكون النتيجة مختلفة تمامًا ، وهي مشكلة في معظم خوارزميات التعلم الآلي. تمتلك معظم خوارزميات التعلم الآلي فهمًا سطحيًا للبيانات. فما هو الحل؟ الإجابة هي إعطاء الجهاز تمثيلاً أكثر تجريدًا للبيانات. ومع ذلك ، بالنسبة للعديد من المهام ، من المستحيل معرفة الخصائص التي يجب استخراجها. هذا هو المكان الذي تدخل فيه فكرة التعلم التمثيلي.

التعلم التمثيلي هو مجموعة فرعية من التعلم الآلي الذي يهدف إلى الحصول على ميزات جيدة ومفيدة للبيانات تلقائيًا ، دون إشراك مهندس ميزة. في هذا النهج ، يأخذ الجهاز البيانات الأولية كمدخلات ويكتشف تلقائيًا التمثيلات اللازمة لتحديد الميزة ، ثم يتعلم تلقائيًا الميزات الجديدة ويطبقها. بعبارة أخرى ، الغرض من التعلم التمثيلي هو العثور على تحويل يرسم البيانات الأولية إلى التمثيل الأكثر ملاءمة لمهمة التعلم الآلي (على سبيل المثال ، التصنيف). نظرًا لأنه يمكن تفسير هذه الطريقة على أنها تعلم ميزات مفيدة ، فإنها تسمى أيضًا تعلم الميزات.

في الأساس ، التعلم التمثيلي ليس أكثر من مجموعة من السمات التي تصف المفاهيم بشكل فردي. على سبيل المثال ، يمكننا تمثيل الكائنات بألوانها وأشكالها وأحجامها وخصائصها. التمثيل هو شيء يساعدنا على التمييز بين المفاهيم المختلفة ويساعدنا بدوره على إيجاد أوجه التشابه بينها.

التعلم االتمثيلى يقلل البيانات عالية الأبعاد إلى بيانات منخفضة الأبعاد ، ويسهل العثور على الأنماط والشذوذ ، كما يمنحنا فهمًا أفضل لسلوك البيانات. مع انخفاض تعقيد البيانات ، يتم تقليل الانحرافات والضوضاء. يمكن أن يكون تقليل الضوضاء مفيدًا جدًا لخوارزميات التعلم الخاضع للإشراف.

النقاط المتطرفة

كما ذكر أعلاه ، فإن جودة البيانات مهمة للغاية. إذا تم تدريب خوارزميات التعلم الآلي على نقاط متطرفة نادرة ومشوهة ، فإن تعميمها يكون أقل مما لو تم تجاهلها بشكل منفصل. من خلال التعلم غير الإشرافي ، يمكننا اكتشاف النقاط المتطرفة باستخدام تقليل الأبعاد وإنشاء حل للبيانات المعيارية.

انحراف البيانات[1]

الفرضية الأساسية لتطوير أي نموذج للتعلم الآلي هي أن البيانات المستخدمة لتعليم النموذج تحاكي بيانات العالم الحقيقي. ولكن كيف يمكن تأكيد هذا الافتراض بعد إنشاء النموذج في الإنتاج؟ عندما تقوم بتدريب نموذج باستخدام نهج التعلم الخاضع للإشراف ، يتم تسمية بيانات التدريب ، وعندما تنشر النموذج في الإنتاج (المعالجة) ، لا توجد تسمية حقيقية ، بغض النظر عن مدى دقة نموذجك ، تكون التنبؤات صحيحة فقط إذا تم تقديم البيانات إلى النموذج في الإنتاج يحاكي (أو يكون مكافئًا إحصائيًا) البيانات المستخدمة في التدريب. ماذا لو لم تفعل؟ في هذه الحالة ، نسميها انحراف البيانات.

يجب أن تكون نماذج التعلم الآلي على دراية بانحراف البيانات. إذا كانت البيانات التي يتنبأ بها النموذج مختلفة إحصائيًا عن البيانات التي يتم تدريب النموذج بناءً عليها ، فقد يحتاج النموذج إلى إعادة التدريب على البيانات الأكثر دراية بالبيانات الحالية. إذا لم يتم إعادة تدريب النموذج أو لم يكتشف الانحراف ، فستتأثر جودة توقع النموذج في البيانات الحالية. من خلال إنشاء توزيعات احتمالية باستخدام التعلم غير الإشرافي ، يمكننا تقييم مدى اختلاف البيانات الحالية عن بيانات مجموعة التدريب. إذا كان الاثنان مختلفين تمامًا عن بعضهما البعض ، فيمكننا البدء في إعادة التدريب تلقائيًا.

التجميع (الكلاستيرينك)

التجميع أو التحليل العنقودي مهمته تجميع مجموعة من الكائنات بحيث تكون الكائنات في مجموعة (تسمى العنقود او الكلاستر) أكثر تشابهًا مع بعضها البعض من المجموعات الأخرى (العناقيد). بتعبير أدق ، الهدف من تحليل الكلاستر (من الناحية المثالية) هو إيجاد مجموعات تكون فيها العينات داخل كل مجموعة متشابهة تمامًا ، بينما تختلف كل مجموعة تمامًا عن بعضها البعض. الكلاستر ليست أكثر من تجميع نقاط البيانات بحيث تكون المسافة بين نقاط البيانات داخل الكلاستر ضئيلة. بمعنى آخر ، المجموعات هي مناطق تكون فيها كثافة نقاط

[1] Data drift

البيانات المتشابهة عالية. بشكل عام ، يتم استخدام التجميع لتحليل مجموعة البيانات من أجل العثور على **بيانات واستدلالات ثاقبة**[1]. أيضًا ، تعتمد الاستنتاجات التي يتم استخلاصها من مجموعة البيانات على المستخدم ، حيث لا توجد معايير محددة بشكل عام للتجميع الجيد.

لإجراء هذا التحليل ، تتلقى خوارزميات التجميع البيانات وتشكل هذه المجموعات (العناقيد) باستخدام بعض معايير التشابه. لتحقيق تجميع ناجح ، يجب أن تحقق خوارزمية التجميع هدفين رئيسيين: (1) ، التشابه بين نقطة بيانات وأخرى ، و (2) ، تشابه نقاط البيانات هذه مع نقاط أخرى تختلف بالتأكيد استكشافيًا عنها. لديك. ومن ثم ، في عملية التجميع ، يلعب معيار التشابه القائم على المسافة دورًا مهمًا في اتخاذ القرارات العنقودية.

باستخدام الهدفين المذكورين أعلاه ، يجب عليك اختيار خوارزمية التجميع التي من المحتمل أن تحصل على أفضل النتائج لمشكلتك الخاصة. تحتوي بعض خوارزميات التجميع على تعريفات مختلفة لماهية الكلاستر. على سبيل المثال ، قد تحدد عدة خوارزميات الكلاستر على أنها مجموعة من البيانات ذات أشكال هندسية محددة. على سبيل المثال ، k-Means ، الذي يعرف الكلاستر على أنها مجموعة من الكائنات ذات شكل كروي. وفي الوقت نفسه ، قد تفترض مجموعة أخرى من الخوارزميات أن المجموعات موجودة بكثافة في مساحة البيانات ، على سبيل المثال DBSCAN. بالإضافة إلى ذلك ، هناك المزيد من التعريفات العنقودية. وفقًا لذلك ، تعتمد جودة النتيجة على الملاءمة بين تعريف الكلاستر المفترض للخوارزميات والبنية العنقودية الجوهرية للبيانات. لذلك ، عند محاولة إجراء هذا التحليل ، من الأفضل أن يكون لديك معرفة مسبقة بتعريف الكلاستر الأكثر ملاءمة لمشكلتك. ومع ذلك ، قد لا تتوفر هذه المعرفة عند بدء عملية التجميع. هذا عادة ما يعقد التحليل.

يمكن استخدام التجميع بمفرده لتحديد البنية الجوهرية للبيانات ، في حين أنه يمكن أيضًا أن يكون بمثابة تقنية معالجة مسبقة لمهام التعلم الأخرى مثل التصنيف. على سبيل المثال ، قد ترغب شركة ما في تصنيف المستخدمين الجدد إلى "مجموعات" مختلفة ، ولكن قد لا يكون ذلك سهلاً. في مثل هذه الحالة ، يمكننا استخدام التجميع لتجميع جميع المستخدمين في مجموعات ، حيث تمثل كل مجموعة فئة مستخدم. بعد ذلك ، يمكن بناء نموذج تصنيف على مجموعات لتصنيف مجموعات مستخدمين جديدة.

[1] insightful data

التجميع وانواعه

بشكل رسمي ، لنفترض أن لدينا مجموعة بيانات $D = \{x_1, x_2, \dots, x_m\}$ تحتوي على عينات غير معنونة ، حيث تكون كل عينة $x_i = (x_{i1}; x_{i2}; \dots, x_{in})$ عبارة عن متجه n أبعاد. وفقًا لذلك، تقسم خوارزمية التجميع مجموعة البيانات D إلى مجموعات منفصلة k

$$\{C_l | l = 1,2, \dots, k\}$$

حيث $C_l \cap_{\acute{l} \neq l} C_{\acute{l}} = \emptyset$ و $D = \cup_{l=1}^{k} C_l$. ومن ثم ، فإننا نعرض $\lambda_j \epsilon \{1,2, \dots, k\}$ به كعينة تسمية المجموعة x_j. أخيرًا ، يمكن تمثيل نتيجة التجميع على أنها تسمية متجه للمجموعة λ $\lambda = (\lambda_1; \lambda_2; \dots; \lambda_m)$ مع عناصر m.

من الناحية المفاهيمية ، يمكن اعتبار التجميع مشابهًا للتصنيف ، من حيث أنه يحاول تعيين قيمة منفصلة لكل حالة. الاختلاف الوحيد هو أنه بينما يستخدم التصنيف عينات مصنفة لتعلم الأنماط في البيانات لفئات منفصلة ، فإن التجميع ليس له معرفة مسبقة بعضوية الفئة أو ما إذا كانت هناك فئات منفصلة في البيانات. ومن ثم ، فإن التجميع يصف مجموعة من الخوارزميات التي تحاول تحديد هيكل التجميع في مجموعة البيانات. باختصار ، **الغرض من التجميع الوصف والغرض من التصنيف التنبؤ**. اعتمادًا على استراتيجية التعلم المستخدمة ، يمكن تقسيم خوارزميات التجميع إلى عدة فئات:

- **التجميع القائم على النموذج الأولي.**
- **التجميع على أساس الكثافة.**
- **المجموعات الهرمية.**

في استمرار لهذا القسم ، سنقوم بوصفهم. قبل ذلك ، دعونا أولاً نناقش مشكلتين أساسيتين في التجميع: **تقييم الأداء وحساب المسافة.**

تقييم الاداء (مؤشرات المصداقية [1])

معايير التقييم الخاصة بالتجميع تسمى أيضًا مؤشرات الصلاحية. نظرًا لأن نتيجة التصنيف يتم تقييمها من خلال معايير أداء التعلم الخاضع للإشراف ، فيجب أيضًا تقييم نتيجة التجميع من خلال بعض مؤشرات الصلاحية. قبل أن نتعمق في ذلك ، فإن تقييم ما إذا كانت مجموعة معينة جيدة أمر إشكالي ومثير للجدل. في الواقع ، كان بونر (1964) أول من جادل بأنه لا يوجد تعريف عام للتجميع الجيد. مزيد من التقييم في عين الناظر. ومع ذلك ، فقد تم تطوير العديد من معايير التقييم. تنقسم هذه المعايير عادة إلى فئتين من **مؤشرات المصداقية الداخلية والخارجية**. يقارن

[1] validity indices

المؤشر الخارجي نتيجة التجميع بنموذج مرجعي ، بينما يقوم المؤشر الداخلي بتقييم نتيجة التجميع دون استخدام أي نموذج مرجعي.

بالنظر إلى مجموعة البيانات $D = \{x_1, x_2, \dots, x_m\}$ ، افترض أن خوارزمية التجميع تنتج عناقيد $c = \{C, C_2, \dots, C_k\}$ و ونموذج مرجعي للمجموعات $c^* = \{C^*, C^*_2, \dots, C^*_k\}$. وفقاً لذلك ، دع λ و λ^* تشير إلى تسميات المجموعات c و c^* ، على التوالي. ومن ثم ، يمكننا تحديد التعبيرات الأربعة التالية لكل زوج من العينات:

$$a = |SS|, SS = \{(\boldsymbol{x_i}, \boldsymbol{x_j}) \mid \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}$$

$$b = |SD|, SD = \{(\boldsymbol{x_i}, \boldsymbol{x_j}) \mid \lambda_i = \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}$$

$$c = |DS|, DS = \{(\boldsymbol{x_i}, \boldsymbol{x_j}) | \lambda_i \neq \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}$$

$$d = |DD|, DD = \{(\boldsymbol{x_i}, \boldsymbol{x_j}) \mid \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}$$

حيث يتكون SS من أزواج من العينات ، كلاهما ينتمي إلى مجموعة في c وأيضًا إلى مجموعة في c^*. تتكون مجموعة SD من أزواج من العينات التي تنتمي إلى نفس المجموعة في c ولكن ليست في c^*. يمكن تفسير مجموعات DD و DS بالمثل. نظرًا لأن كل زوج من العينات (x_i, x_j) $(i < j)$ يمكن أن يظهر في مجموعة واحدة فقط ، فلدينا:

$$a + b + c + d = \frac{m(m-1)}{2}$$

وفقاً للمصطلحات الأربعة المحددة أعلاه ، يمكن تعريف بعض المؤشرات الخارجية المشتركة على النحو التالي:

- **معامل جاكارد [1] (JS):**

$$JS = \frac{a}{a+b+c}$$

- **مؤشر فولكس ومالوس [2] (FMI):**

$$FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$$

- **مؤشر راند (RI)[3]:**

$$RI = \frac{2(a+d)}{m(m-1)}$$

---

[1] Jaccard Coefficient

[2] Fowlkes and Mallows Index

[3] Rand Index

تأخذ مؤشرات المصداقية الخارجية المذكورة أعلاه قيمًافي النطاق [1، 0]. كلما زادت قيمة المؤشر، كانت جودة التجميع أفضل.

تعمل مؤشرات الصلاحية الداخلية على تقييم جودة التجميع بدون استخدام النموذج المرجعي. بالنظر إلى المجموعات التي تم إنشاؤها $c = \{C, C_2, \ldots, C_k\}$، يمكننا تحديد المصطلحات الأربعة التالية:

$$avg(C) = \frac{2}{|C|(|C|-1)} \sum_{1 \le i < j \le |C|} dist(x_i, x_j)$$
$$diam(C) = max_{1 \le i < j \le |C|} dist(x_i, x_j)$$
$$d_{min}(C_i, C_j) = min_{x_i \epsilon C_i, x_j \epsilon C_j} dist(x_i, x_j)$$
$$d_{cen}(C_i, C_j) = dist(\mu_i, \mu_j)$$

حيث $\mu = \frac{1}{|C|} \sum_{1 \le i \le |c|} x_i$ يمثل مركز المجموعة C و $dist(.,.)$ يقيس المسافة بين عينتين. هنا، $avg(C)$ هو متوسط المسافة بين العيناتفي المجموعة C. $diam(C)$ هو أكبر مسافة بين العيناتفي المجموعة C. $d_{min}(C_i, C_j)$ هي المسافة بين أقرب عينتينفي المجموعات C_iو C_j و $d_{cen}(C_i, C_j)$ هي المسافة بين مراكز المجموعات C_i و C_j.

وفقًا للمصطلحات المحددة أعلاه ، يمكن تعريف بعض مؤشرات المصداقية الداخلية الشائعة على النحو التالي:

- **مؤشر ديفيز بولدين[1] (DBI):**

$$DBI = \frac{1}{k} \sum_{i=1}^{k} max_{i \ne j} \left(\frac{avg(C_i) + avg(C_j)}{d_{cen}(C_i, C_j)} \right).$$

- **مؤشر دن[2] (DI):**

$$DI = min_{1 \le i \le k} \left\{ min_{i \ne j} \left(\frac{d_{min}(C_i, C_j)}{max_{1 \le l \le k} diam(C_l)} \right) \right\}$$

تشير القيمة المنخفضة لـ DBI إلى جودة تجميع أفضل،في المقابل، تشير القيمة الأكبر لـ DI إلى جودة تجميع أفضل.

---

[1] Davies–Bouldin Index

[2] Dunn Index

حساب المسافة وحساب التشابه

نظرًا لأن تجميع مجموعات العينات / الكائنات متشابه ، هناك حاجة إلى نوع من المعايير لتكون قادرًا على تحديد ما إذا كان كائنان متشابهين أو غير متماثلين. يتم استخدام نوعين رئيسيين من المعايير لتقدير هذه العلاقة: **معيار المسافة ومعيار التشابه.**

تستخدم العديد من طرق التجميع معايير المسافة لتحديد التشابه أو الاختلاف بين كل زوج من الكائنات. يُطلق على معيار المسافة $dist(.,.)$ معيار المسافة المترية إذا كان يفي بالخصائص التالية:

1. **غير سلبي**[1]: $dist(x_i, x_j) \geq 0$
2. **انعكاسى**[2]: $dist(x_i, x_j) = 0 \Leftrightarrow x_i = x_j$
3. **تناظر**[3]: $dist(x_i, x_j) = dist(x_j, x_i)$
4. **متباينة المثلث**[4]: $dist(x_i, x_j) \leq dist(x_i, x_k) + dist(x_k, x_j)$

غالبًا ما نحدد معايير التشابه من خلال بعض أنواع المسافات ، وكلما زادت المسافة ، قل التشابه.

مينكوفسكي: معيار المسافة للخصائص العددية

بالنظر إلى العيّنتين التاليتين $n-$ ابعاد $x_i = (x_{i1}; x_{i2}; \dots, x_{in})$ و $x_j = (x_{j1}; x_{j2}; \dots, x_{jn})$، تم الحصول على المسافة بين العينتين باستخدام مقياس مينكوفسكي على النحو التالي:

$$dist(x_i, x_j) = \left(|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + \cdots + |x_{ip} - x_{jp}|^g\right)^{\frac{1}{g}}$$

لاحظ أنه إذا كانت $g = 2$، 1 و ∞ نحصل على المسافة الإقليدية ، ومسافة مانهاتن ، والحد الأقصى للمسافة ، على التوالي.

يمكن أن تؤثر وحدة القياس المستخدمة على تحليل المجموعات. لتجنب الاعتماد على اختيار وحدة القياس ، يجب مزامنة البيانات (قياسها). المعيار المطابق يحاول إعطاء كل المتغيرات نفس الأهمية. ومع ذلك ، إذا تم ترجيح كل متغير وفقًا لأهميته ، فيمكن حساب المسافة الموزونة على النحو التالي:

---

[1] Non-negativity

[2] reflexivity

[3] Symmetry

[4] triangle inequality

$$dist(x_i, x_j) = \left(w_1|x_{i1} - x_{j1}|^g + w_2|x_{i2} - x_{j2}|^g + \cdots + w_i|x_{ip} - x_{jp}|^g\right)^{\frac{1}{g}}$$

جایی که $w_i \epsilon [0, \infty)$ است.

معيار المسافة للخصائص الثنائية

في حالة الخصائص الثنائية ، يمكن حساب المسافة بين الأشياء بناءً على **جدول التمرير**[1]. تكون الخاصية الثنائية متماثلة إذا كان لكلا الوضعين نفس القيمة. في هذه الحالة ، باستخدام **معامل المطابقة البسيط**[2] ، يمكن تقييم الاختلاف بين الكائنين:

$$dist(x_i, x_j) = \frac{r + s}{q + r + s + t}$$

حيث q هو عدد الخصائص التي تساوي 1 لكلا الكائنين ، t هو عدد الخصائص التي تساوي 0 لكلا الكائنين ، r و s و هي عدد الخصائص غير المتكافئة لكلا الكائنين.

معيار المسافة للخصائص الاسمية

عندما تكون السمات اسمية ، يمكن استخدام نهجين رئيسيين:

1. تطابق بسيط:

$$dist(x_i, x_j) = \frac{p - m}{p}$$

حيث p هو العدد الإجمالي للخصائص و m هو عدد التطابقات.

2. إنشاء سمة ثنائية لكل حالة من كل ميزة اسمية وحساب الاختلاف بينها.

التجميع القائم على النموذج الأولي[3]

تجميع النموذج الأولي ، المعروف أيضًا باسم التجميع المستند إلى النموذج الأولي ، هو عائلة من خوارزميات التجميع التي تفترض أنه يمكن تمثيل بنية التجميع بمجموعة من النماذج الأولية. عادةً ما تبدأ هذه الخوارزميات ببعض النماذج الأولية ثم تقوم بتحديث النماذج الأولية وتحسينها بشكل متكرر. تم تطوير العديد من الخوارزميات باستخدام نهج النموذج الأولي وطرق التحسين المختلفة. ومع ذلك ، في استمرار هذا القسم ، سنناقش مثالين فقط من خوارزميات التجميع المستندة إلى النموذج الأولي ، وهما مجموعات **k-Means** و**التجميع المختلط Gaussian**.

---

[1] contingency table

[2] simple matching coefficient

[3] Prototype clustering

التجميع k-Means

تقنية التجميع K-mean بسيطة ونبدأ بشرح الخوارزمية الرئيسية. نختار أولاً K المركز الأولي، حيث K هي معامل يحدده المستخدم ، أي عدد المجموعات المطلوبة. ثم يتم تعيين كل نقطة إلى أقرب مركز ، وكل مجموعة من النقاط المخصصة للمركز هي مجموعة. ثم يتم تحديث مركز كل عنقود بناءً على النقاط المخصصة للمجموعة. نكرر خطوات التعيين والتحديث حتى لا تتغير أي نقاط في المجموعات ، بمعنى آخر ، طالما ظلت المراكز ثابتة.

ضع في اعتبارك البيانات التي معيار التشابه هو المسافة الإقليدية. بالنسبة للدالة الموضوعية ، التي تقيس جودة التجميع ، نستخدم **تقليل الخطأ التربيعي**[1]. بمعنى آخر ، نحسب أولاً خطأ كل نقطة بيانات (أي المسافة الإقليدية لأقرب مركز) ثم مجموع مربعات الأخطاء. بالنظر إلى مجموعتين مختلفتين من المجموعات التي تم إنشاؤها بواسطة تطبيقين مختلفين للوسائل K ، فإننا نفضل المجموعة التي بها أخطاء المربعات الصغرى. لأنه يعني أن النماذج الأولية (الوسط) لهذه المجموعة تمثل تمثيلًا أفضل للنقاط الموجودة في مجموعتها.

ببشكل رسمي ، بالنظر إلى مجموعة البيانات $D = \{x_1, x_2, \dots, x_m\}$، تقلل خوارزمية k-mean الخطأ التربيعي للمجموعات $c = \{C, C_2, \dots, C_k\}$:

$$E = \sum_{i=1}^{k} \sum_{x \epsilon C_i} \|x - \mu_i\|_2^2$$

حيث $\mu_i = \frac{1}{|C_i|} \sum_{x \epsilon C_i}$ و متوسط متجه مجموعةC_i. بشكل حدسي ، تُظهر المعادلة أعلاه التقارب (الجوار) بين متوسط متجه المجموعة والعينات داخل تلك المجموعة ، حيث يشير الحرف E الأصغر إلى التشابه داخل الكتلة الأعلى. ومع ذلك ، فإن تقليل E ليس بالأمر السهل، لأنه يتطلب تقييم جميع الأقسام الممكنة لمجموعة بيانات D ، والتي تعد في الواقع **مساله NP-hard**. ومن ثم ، فإن خوارزمية **k-mean** تتبنى استراتيجية جشعة وتعتمد طريقة تحسين تكرارية للعثور على الحل التقريبي. في هذه الخوارزمية ، أولاً ، يتم تهيئة المتجهات المتوسطة ويتم تحديث المجموعات والمتجهات المتوسطة بشكل متكرر ، على التوالي. عندما لا تتغير المجموعات بعد تكرار واحد ، يتم إرجاع المجموعات الحالية.

فيما يلي مثال لفهم أفضل. ومع ذلك ، فإننا نعرض الخوارزمية في حالة لا يوجد فيها سوى متغيرين فقط بحيث يمكن تمثيل نقاط البيانات ومراكز المجموعة هندسيًا بنقاط على مستوى إحداثي. نحسب المسافة بين النقطتين (x_1, x_2) و (y_1, y_2) باستخدام صيغة المسافة الإقليدية:

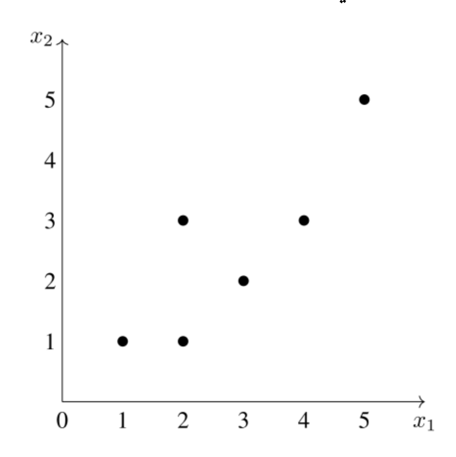
[1] minimizes the squared error

$$||\vec{x} - \vec{y}|| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

مثال. باستخدام خوارزمية نقاط البيانات ، نريد تقسيم الجدول التالي إلى مجموعتين:

x_1	1	2	2	3	4	5
x_2	1	1	3	2	3	5

يمكن رؤية مخطط مبعثر البيانات في الشكل أدناه:



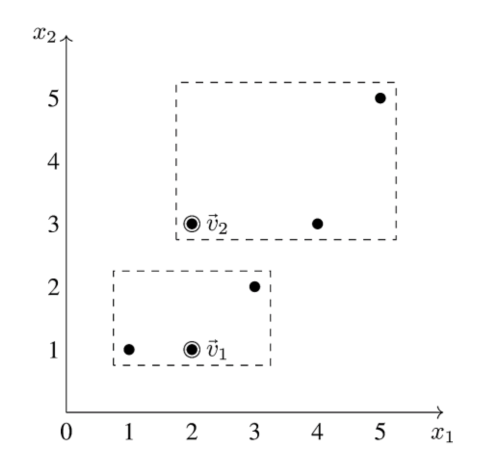
الآن سوف نصف الخوارزمية خطوة بخطوة:

1. في المشكلة ، العدد المطلوب من العناقيد هو 2. ومن ثم ، قمنا بتعيين $k = 2$.
2. نختار نقطتين كمراكز المجموعة الأولية. دعنا نختار $\vec{v}_1 = (2,1)$ و $\vec{v}_2 = (2,3)$.
3. فنحسب نقاط البيانات الأصلية من مراكز المجموعة:

$\vec{x}_i$	نقاط البيانات	المسافة من $\vec{v}_1 = (2,1)$	المسافة من $\vec{v}_2 = (2,3)$	أقصر مسافة	المراكز المعينة
$\vec{x}_1$	(1,1)	1	2.24	1	$\vec{v}_1$
$\vec{x}_2$	(2,1)	0	2	0	$\vec{v}_1$
$\vec{x}_3$	(2,3)	2	0	0	$\vec{v}_2$
$\vec{x}_4$	(3,2)	1.41	1.41	0	$\vec{v}_1$
$\vec{x}_5$	(4,3)	2.82	3	2	$\vec{v}_2$
$\vec{x}_6$	(5,5)	5	3.61	3.61	$\vec{v}_2$

(المسافة $\vec{x}_4$ از $\vec{v}_1$ و $\vec{v}_2$ متساوية. قمنا بشكل اختياري بتعيين $\vec{v}_1$ را به $\vec{x}_4$)

بناءً على ذلك، يتم تقسيم البيانات إلى مجموعتين على النحو التالي:



المجموعة1: تم تعيين $\{\vec{x}_1,\vec{x}_2,\vec{x}_4\}$ به $\vec{v}_1$.

عدد نقاط البيانات في المجموعة1: $C_1 = 3$

المجموعة 2: تم تعيين $\{\vec{x}_3,\vec{x}_5,\vec{x}_6\}$ به $\vec{v}_2$.

عدد نقاط البيانات في المجموعة 2: $C_2 = 3$

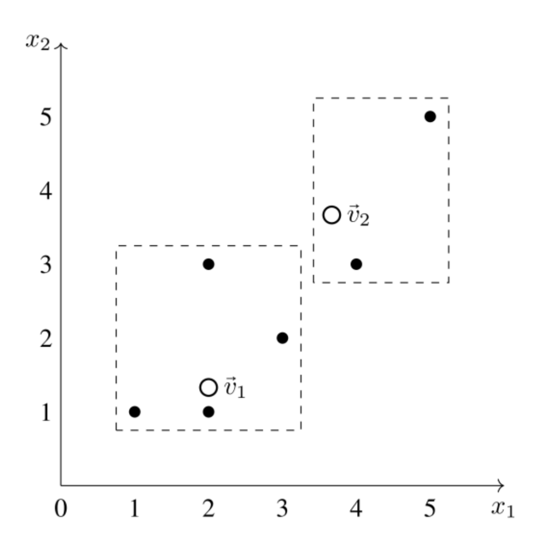
4. يتم إعادة حساب مراكز المجموعة على النحو التالي:

$$\vec{v}_1 = \frac{1}{C_1}(\vec{x}_1 + \vec{x}_2 + \vec{x}_4)$$
$$= \frac{1}{3}(\vec{x}_1 + \vec{x}_2 + \vec{x}_4)$$
$$= (2.00,1.33)$$
$$\vec{v}_2 = \frac{1}{C_2}(\vec{x}_3 + \vec{x}_5 + \vec{x}_6)$$
$$= \frac{1}{3}(\vec{x}_3 + \vec{x}_5 + \vec{x}_6)$$
$$= (3.67,3.67)$$

5. نحسب مسافة نقاط البيانات المعطاة من مراكز الكتلة الجديدة:

$\vec{x}_i$	نقاط البيانات	المسافة من $\vec{v}_1 =$ (2, 1.33)	المسافة من $\vec{v}_2 =$ (3.67, 3.67)	اقصر مسافة	المراكز المعينة
$\vec{x}_1$	(1,1)	1.05	3.77	1.05	$\vec{v}_1$
$\vec{x}_2$	(2,1)	0.33	3.14	0.33	$\vec{v}_1$
$\vec{x}_3$	(2,3)	1.67	1.80	1.67	$\vec{v}_1$
$\vec{x}_4$	(3,2)	1.20	1.80	1.20	$\vec{v}_1$
$\vec{x}_5$	(4,3)	2.60	0.75	0.75	$\vec{v}_2$
$\vec{x}_6$	(5,5)	4.74	1.89	1.89	$\vec{v}_2$

بناءً على ذلك ، يتم تقسيم البيانات إلى مجموعتين على النحو التالي:



المجموعة 1: تم تعيين $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4\}$ به $\vec{v}_1$.

عدد نقاط البيانات في المجموعة 1: $C_1 = 4$

المجموعة 2: تم تعيين $\{\vec{x}_5, \vec{x}_6\}$ به $\vec{v}_2$.

عدد نقاط البيانات في المجموعة 2: $C_2 = 2$

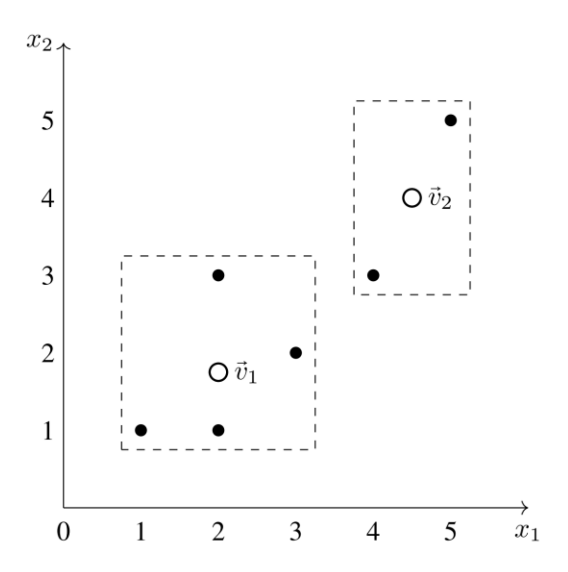
6. يتم إعادة حساب مراكز المجموعة على النحو التالي:

$$\vec{v}_1 = \frac{1}{C_1}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \vec{x}_4)$$
$$= \frac{1}{4}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \vec{x}_4)$$
$$= (2.00, 1.75)$$
$$\vec{v}_2 = \frac{1}{C_2}(\vec{x}_5 + \vec{x}_6)$$
$$= \frac{1}{2}(\vec{x}_5 + \vec{x}_6)$$
$$= (4.5, 4)$$

7. نحسب مسافة نقاط البيانات المعطاة من مراكز المجموعة الجديدة:

$\vec{x}_i$	نقاط داده	فاصله از $\vec{v}_1$	فاصله از $\vec{v}_2$	كمترين فاصله	مركز اختصاص داده شده
$\vec{x}_1$	(1,1)	1.25	4.61	1.25	$\vec{v}_1$
$\vec{x}_2$	(2,1)	0.75	3.91	0.75	$\vec{v}_1$
$\vec{x}_3$	(2,3)	1.25	2.69	1.25	$\vec{v}_1$
$\vec{x}_4$	(3,2)	1.03	2.50	1.03	$\vec{v}_1$
$\vec{x}_5$	(4,3)	2.36	1.12	1.12	$\vec{v}_2$
$\vec{x}_6$	(5,5)	4.42	1.12	1.12	$\vec{v}_2$

بناءً على ذلك ، يتم تقسيم البيانات إلى مجموعتين على النحو التالي:



المجموعة 1: تم تعيين $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4\}$ به $\vec{v}_1$.

عدد نقاط البيانات في المجموعة 1: $C_1 = 4$

خوشه 2: تم تعيين $\{\vec{x}_5, \vec{x}_6\}$ به $\vec{v}_2$.

عدد نقاط البيانات في المجموعة 2: $C_2 = 2$

8. يتم إعادة حساب مراكز المجموعة على النحو التالي:

$$\vec{v}_1 = \frac{1}{C_1}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \vec{x}_4)$$
$$= \frac{1}{4}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \vec{x}_4)$$
$$= (2.00, 1.75)$$
$$\vec{v}_2 = \frac{1}{C_2}(\vec{x}_5 + \vec{x}_6)$$
$$= \frac{1}{2}(\vec{x}_5 + \vec{x}_6)$$
$$= (4.5, 4)$$

9. نظرًا لأن هذه هي نفس مراكز المجموعات المحسوبة مسبقًا ، فلن يكون هناك إعادة توزيع لنقاط البيانات إلى مجموعات أخرى ، وبالتالي تتوقف الحسابات هنا.

10. نتيجة لذلك ، يتم الحصول على مجموعات من المراكز التالية على النحو التالي:

المجموعة 1: $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4\}$ به $\vec{v}_1 = (2, 1.75)$.

المجموعة 2: $\{\vec{x}_5, \vec{x}_6\}$ به $\vec{v}_2 = (4.5, 4)$.

مميزات التجميع k-means

- لها تعقيد زمني خطي. ومن ثم فهو سريع وفعال.
- إرجاع المجموعات التي يسهل تفسيرها وحتى تصورها. يمكن أن تكون هذه البساطة مفيدة للغاية في بعض الحالات حيث تحتاج إلى نظرة عامة سريعة على أقسام البيانات.
- سهلة التنفيذ.
- تعمل بشكل رائع عند التعامل مع العناقيد الكروية.

معايب خوشه‌بندی k-means

- تحتاج خوارزمية التعلم إلى تحديد عدد مراكز المجموعات مقدمًا.
- تعتمد مراكز المجموعة النهائية على المراكز الأولية المختارة. بمعنى آخر ، تؤدي التقسيمات الأولية المختلفة إلى مجموعات نهائية مختلفة.
- حساسة للمواقع البعيدة.

التعقيد المكاني والزماني

المساحة المطلوبة للوسائل K ليست كبيرة ، حيث يتم تخزين نقاط البيانات والمراكز فقط. بشكل عام ، التخزين المطلوب هو $O((m + K)n)$ ، حيث m هو عدد النقاط و n هو عدد السمات. الوقت المطلوب للوسائل K خطي من حيث عدد نقاط البيانات. بشكل عام ، الوقت المطلوب هو $O(I \times K \times m \times n)$، حيث I هو عدد التكرارات المطلوبة للتقارب. غالبًا ما أكون صغيرًا ويمكن عادةً تقييده بأمان ، حيث تحدث معظم التغييرات عادةً خلال التكرارات القليلة الأولى. لذلك ، فإن K-means من حيث m (عدد النقاط) خطي وفعال وبسيط؛ بشرط أن يكون K (عدد المجموعات) أقل بكثير من m.

التجميع الغاوسي المختلط

في القسم السابق ، قدمنا المجموعات k ، وهو أحد أكثر طرق التجميع استخدامًا. تعمل هذه الطريقة غير البارامترية بشكل رائع لتجميع البيانات مع بعض الميزات. في هذا القسم ، سنفحص طريقة بارامترية تستخدم التوزيع الغاوسي ، والمعروفة باسم **نموذج غاوسي المختلط**. تتمثل إحدى نقاط القوة في تجميع نموذج غاوسي المختلط في أنه طريقة تجميع ناعمة. بعبارات أبسط، فإنه يناسب مجموعة من النماذج الممكنة مع البيانات ويحدد احتمال الانتماء إلى كل حالة. هذا يسمح لنا بفحص احتمال أن كل عنصر ينتمي إلى كل مجموعة. وبالتالي ، فإن التجميع النموذجي المختلط يناسب مجموعة من النماذج الممكنة مع البيانات. يمكن أن تكون هذه النماذج مجموعة متنوعة من التوزيعات الممكنة ، ولكنها عادة ما تكون توزيعات غاوسية. يُطلق على نهج التجميع هذا اسم النمذجة المختلطة لأننا نلائم توزيعات متعددة محتملة (مخاليط) للبيانات. وبالتالي ، فإن النموذج الغاوسي المختلط هو ببساطة نموذج يناسب توزيعات غاوسي المتعددة في مجموعة

من البيانات. يمثل كل غاوسي في النموذج المختلط مجموعة محتملة. بمجرد أن يكون مزيج غاوسي الخاص بنا متسقًا مع البيانات قدر الإمكان ، يمكننا حساب احتمال أن كل عنصر ينتمي إلى كل مجموعة وتعيين العناصر إلى المجموعة الأكثر احتمالية. السؤال الذي يطرح نفسه هنا هو أنه ليس لدينا معرفة بالتوزيع الذي تم اشتقاق كل عينة تدريب منه ، ولا معاملات النموذج المختلط ، ومع ذلك ، كيف يمكننا العثور على مجموعة من غاوسي تتطابق جيداً مع البيانات الأساسية؟ يمكننا اعتماد وتكرار الطريقة المستخدمة في خوارزمية التجميع k-means. وبالتالي، نبدأ بالتخمينات الأولية للمعاملات ونستخدمها لحساب احتمالات المجموعة لكل حالة. بعد ذلك ، نستخدم هذه الاحتمالات لإعادة تقدير المعاملات وتكرار هذه الدورة حتى التقارب. يتم ذلك عن طريق خوارزمية تسمى **تعظيم التوقع**[1].

في ظل ظروف معينة ، يمكن تفسير k-means ونموذج الخليط الغاوسي ببعضهما البعض. في k-means ، يتم تعيين النقاط الأقرب إلى مركز المجموعة مباشرة إلى مركز المجموعة هذا ، بافتراض أن المجموعات يتم تحجيمها بشكل مشابه وأن تباين المميزات لا يختلف. هذا هو السبب في أنه غالبًا ما يكون من المنطقي توحيد بياناتك قبل استخدام k-means. ومع ذلك ، لا يعاني خليط غاوسي من مثل هذا القيد ، لأنهم يسعون إلى نمذجة التغاير المحدد لكل عنقود. بعبارة أخرى ، على عكس k-means ، لا تستخدم مجموعات الخليط الغاوسي نواقل النموذج الأولي، ولكنها تستخدم النماذج الممكنة لتمثيل هياكل العنقودية. تفترض النماذج المختلطة من غاوسيان أن كل ملاحظة في مجموعة بيانات تختلف عن توزيع غاوسيان بمتوسط وتباين. من خلال ملاءمة البيانات للنموذج المختلط غاوسي ، هدفنا هو تقدير معاملات التوزيع الغاوسي باستخدام البيانات. إذا كان هذا يبدو محيرا بعض الشيء ، فلا تقلق! سنراجع هذه المفاهيم بمزيد من التفصيل. ومع ذلك ، قبل مناقشة هذه الخوارزمية بمزيد من التفصيل ، دعنا نراجع تعريف التوزيع الغاوسي (متعدد المتغيرات).

ربما سمعت عن التوزيع الغاوسي ، الذي يُعرف أحيانًا بالتوزيع الطبيعي ، ولكن ما هو التوزيع الغاوسي بالضبط؟ سنقدم لك التعريف الرياضي قريبًا ، ولكن من الناحية النوعية يمكن اعتباره توزيعًا يحدث بشكل طبيعي ومتكرر جداً.

بالنسبة للمتجه العشوائي x في فضاء عينة n ذات أبعاد χ ، إذا كان x يتبع التوزيع غاوسيان، فإن دالة كثافة الاحتمال الخاصة به تساوي:

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}}$$

[1] expectation-maximization

حيث μ متجه متوسط الأبعاد n و عبارة عن مصفوفة تغاير $n \times n$. من المعادلة أعلاه ، يمكننا أن نرى أن التوزيع الغوسي محدد تمامًا بواسطة المتجه المتوسط μ ومصفوفة التغاير الخاصة به Σ. لإظهار هذا الاعتماد بشكل أكثر وضوحًا ، نكتب دالة كثافة الاحتمال كـ $p(x \mid \mu, \Sigma)$.

يتم تعريف توزيع المخاليط الغاوسية على النحو التالي:

$$p_{\mathcal{M}}(x) = \sum_{i=1}^{k} \alpha_i . p(x \mid \mu, \Sigma)$$

والذي يتكون من مكونات k للخليط ، كل منها مرتبط بتوزيع غاوسيان . μ_i و Σ_i هي معاملات مكونات الخليطi ام، $\alpha_i > 0$ هي معاملات الخليط المقابلة حيث $\alpha_i > 0$. افترض أن العينات تم إنشاؤها من توزيع خليط غاوسي بالعملية التالية:

حدد مكون الخليط الغاوسي باستخدام التوزيع المحدد مسبقًا المحدد بواسطة $\alpha_1, \alpha_1, \ldots, \alpha_k$ حيث α_i هو احتمال اختيار مكون خليط iام. ثم يقوم بتوليد العينات عن طريق أخذ عينات من دوال الكثافة الاحتمالية لمكون الخليط المحدد.

لنفترض أن $D = \{x_1, x_2, \ldots, x_m\}$ هي مجموعة تدريبية تم إنشاؤها بواسطة العملية المذكورة أعلاه و $z_j \epsilon \{1,2, \ldots, k\}$ هي متغير عشوائي لمكون الخليط الغاوسي الذي ينتج العينة x_j في مكان ما. قيم z_j غير معروفة. بما أن الاحتمال السابق $P(z_j = i)$ لـ z_j يتوافق مع $\alpha_i \epsilon \{i = 1,2, \ldots, k\}$، التوزيع اللاحق z_j ، وفقًا لنظرية بايز ، يساوي:

$$p_{\mathcal{M}}(z_j = i | x_j) = \frac{\alpha_i . p_{\mathcal{M}}(x_j | z_j = i)}{p_{\mathcal{M}}(x_j)}$$

$$= \frac{P(z_j = i) . P(x_j | \mu_i, \Sigma_i)}{\sum_{l=1}^{k} \alpha_l . P(x_j | \mu_l, \Sigma_l)}$$

بمعنى آخر ، يعطي $p_{\mathcal{M}}(z_j = i | x_j)$ الاحتمال الأخير بأن x_j يتم إنتاجه بواسطة مكون خليط غاوسي i. للسهولة ، نشير إليه بواسطة γ_{ji} ، حيث $i = 1,2, \ldots, k$.

بمجرد معرفة توزيع خليط غاوسي ، يمكن تقسيم مجموعة D إلى مجموعات k ويتم إعطاء تخصيص المجموعة λ_j لكل عينةx_j بواسطة $\lambda_j = argmax_{i \epsilon \{1,2,\ldots,k\}} \gamma_{ji}$. **وبالتالي ، من منظور تجميع النموذج الأولي ، تستخدم مجموعة خليط غاوسي نماذج احتمالية (مع توزيع غاوسي) لتمثيل النماذج الأولية ، ويتم تخصيص المجموعات بواسطة الاحتمالات اللاحقة للنماذج الأولية.**

الآن السؤال هو كيفية تحسين معاملات النموذج؟ تتمثل إحدى طرق تطبيق تقدير الاحتمالية القصوى على مجموعة بيانات D في تعظيم (log) التحقق:

$$LL(D) = \ln\left(\prod_{j=1}^{m} p_{\mathcal{M}}(x_j)\right)$$
$$= \sum_{j=1}^{m} \ln\left(\sum_{i=1}^{k} \alpha_i . P(x_j|\mu_i, \Sigma_i)\right)$$

والتي عادة ما يتم حلها عن طريق خوارزمية تعظيم التوقعات (EM).

تتكون خوارزمية EM (كما يوحي اسمها) من مرحلتين: **التوقع والتعظيم**. مرحلة الانتظار هي حيث يتم حساب الاحتمالات اللاحقة لكل حالة لغاوسيان(الشكل (ب) 8ـ1 لغاوسيان أحادي البعد والشكل 8ـ2 (b) لغاوسيان أكبر من بعد واحد). في هذه المرحلة ، تستخدم الخوارزمية نظرية بايز لحساب الاحتمالات اللاحقة. الخطوة التالية هي التعظيم. تتمثل مهمة خطوة التعظيم في تحديث معاملات النموذج المختلط لتعظيم صحة البيانات. استنادًا إلى نظرية بايز و $\gamma_{ji} = p_{\mathcal{M}}(z_j = i|x_j)$ ، يتم إجراء تحديث المتوسط على النحو التالي:

$$\mu_i = \frac{\sum_{j=1}^{m} \gamma_{ji} x_j}{\sum_{j=1}^{m} \gamma_{ji}}$$

بمعنى آخر ، يمكن حساب المتوسط (الذي يشير إلى مركز المجموعة) لكل غاوسي على أنه متوسط الاوزان للعينات ، حيث يتم وزن كل عينة مع الاحتمال الأخير للانتماء إلى غاوسي المحدد. يحدث التغاير (يحدد العرض) لكل غاوسيان بطريقة مماثلة:

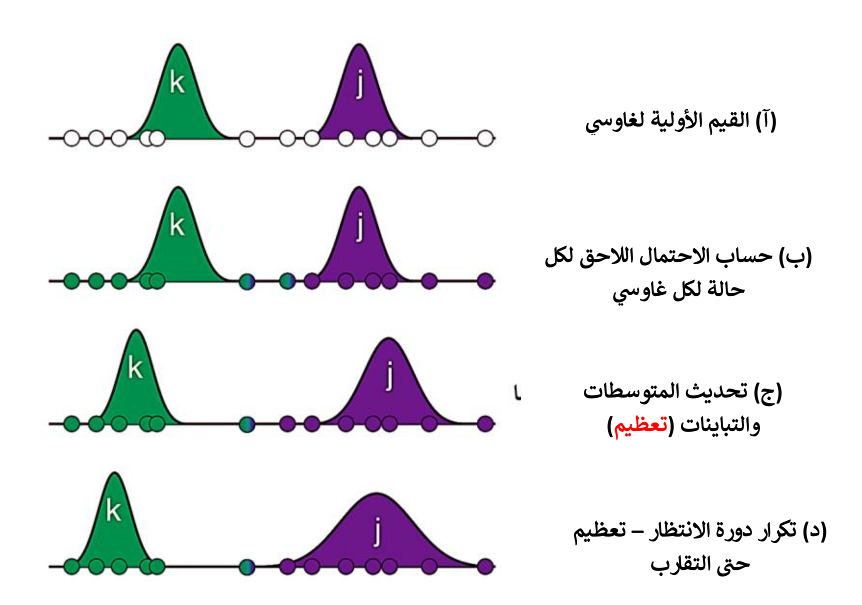
$$\Sigma_i = \frac{\sum_{j=1}^{m} \gamma_{ji}(x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{m} \gamma_{ji}}$$

آخر شيء يجب تحديثه هو الاحتمالات السابقة لكل غاوسي. يتم حساب السجلات الجديدة (معامل الخليط) عن طريق جمع الاحتمالات اللاحقة لغاوسيان معين والقسمة على عدد العينات:

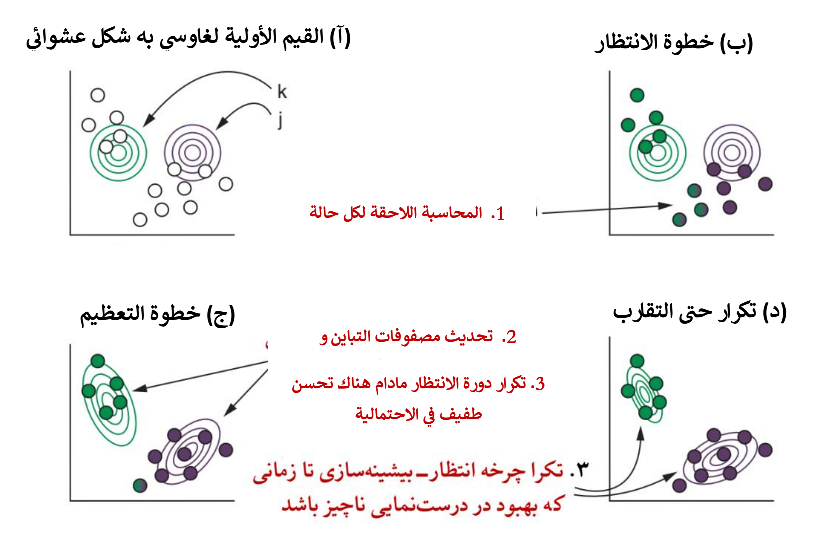
$$\alpha_i = \frac{\sum_{j=1}^{m} \gamma_{ji}}{m}$$

جدول 8ـ1 قاعدة بيانات البطيخ (*Zhou, 2021*)

ID	density	sugar	ID	density	sugar	ID	density	sugar
1	0.697	0.460	11	0.245	0.057	21	0.748	0.232
2	0.774	0.376	12	0.343	0.099	22	0.714	0.346
3	0.634	0.264	13	0.639	0.161	23	0.483	0.312
4	0.608	0.318	14	0.657	0.198	24	0.478	0.437
5	0.556	0.215	15	0.360	0.370	25	0.525	0.369
6	0.403	0.237	16	0.593	0.042	26	0.751	0.489
7	0.481	0.149	17	0.719	0.103	27	0.532	0.472
8	0.437	0.211	18	0.359	0.188	28	0.473	0.376
9	0.666	0.091	19	0.339	0.241	29	0.725	0.445
10	0.243	0.267	20	0.282	0.257	30	0.446	0.459



الشكل 8ـ1. خوارزمية تعظيم الانتظار لاثنين من غاوسي ذات البعد الواحد. تمثل النقاط العناصر الموجودة على طول خط الأعداد. يتم تعيين اثنين من غاوسي بشكل عشوائي على طول خط التهيئة. في مرحلة الانتظار، يتم حساب الاحتمال اللاحق لكل عنصر لكل غاوس ، وفي مرحلة التعظيم ، المتوسطات ، التباينات (في هذا المثال، نظرًا لوجودها في بُعد واحد ، يتم استخدام التباين بدلاً من التباين المشترك) والاحتمالات السابقة لكل غاوسي يتم حسابها على أساس الامتدادات المحسوبة. تستمر هذه العملية حتى يتقارب الاحتمال.



الشكل 8ـ2. خوارزمية تعظيم الانتظار لغاوسي ثنائي الأبعاد. يتم تهيئة اثنين من غاوسي بشكل عشوائي في مساحة الميزة. في مرحلة الانتظار ، يتم حساب الاحتمالات اللاحقة لكل حالة لكل غاوس. في مرحلة التعظيم ، يتم تحديث المتوسطات ومصفوفات التغاير والخلفيات لكل غاوسي بناءً على الامتدادات. تستمر هذه العملية حتى يتقارب الاحتمال.

عند اكتمال خطوة التعظيم ، نقوم بتكرار آخر لخطوة الانتظار ، هذه المرة بحساب الاحتمالات اللاحقة لكل عنصر تحت غاوسيان الجديد. عند الانتهاء من ذلك ، نقوم بتشغيل خطوة التعظيم مرة أخرى ونقوم بتحديث المتوسطات والتفاوتات ومعاملات الخلط (الخلفيات) لكل غاوسي بناءً على ما يلي. تستمر دورة تعظيم التوقع هذه بشكل متكرر حتى تصل إلى عدد معين من التكرارات أو تتغير الصلاحية الإجمالية للبيانات تحت النموذج إلى ما دون قيمة معينة (التقارب).

ضع في اعتبارك مجموعة بيانات البطيخ في الجدول 8ـ1 كمثال لإعطاء رؤية أكثر دقة لتجميع نموذج غاوسي المختلط. افترض أن عدد مكونات خليط غاوسي هو $k = 3$ وأن الخوارزمية تبدأ بتهيئة المعامل التالي:

$$\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3};\ \mu_1 = x_6; \mu_2 = x_{22}; \mu_3 = x_{27};$$

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}$$

في التكرار الأول ، تحسب الخوارزمية الاحتمالات اللاحقة للعينات وفقًا لحقيقة أنها تم إنشاؤها بواسطة كل مكون مختلط. على سبيل المثال ، بالنظر إلى x_1 ، فإن الاحتمالات اللاحقة المحسوبة بواسطة γ_{ji} هي كما يلي:

$$\gamma_{11} = 0.219;\ \gamma_{12} = 0.404;\ \gamma_{13} = 0.377$$

بعد حساب الاحتمالات اللاحقة لجميع العينات فيما يتعلق بجميع مكونات الخليط ، نحصل على معاملات النموذج المحدثة التالية:

$$\acute{\alpha}_1 = 0.361; \acute{\alpha}_2 = 0.323;\ \acute{\alpha}_3 = 0.316$$

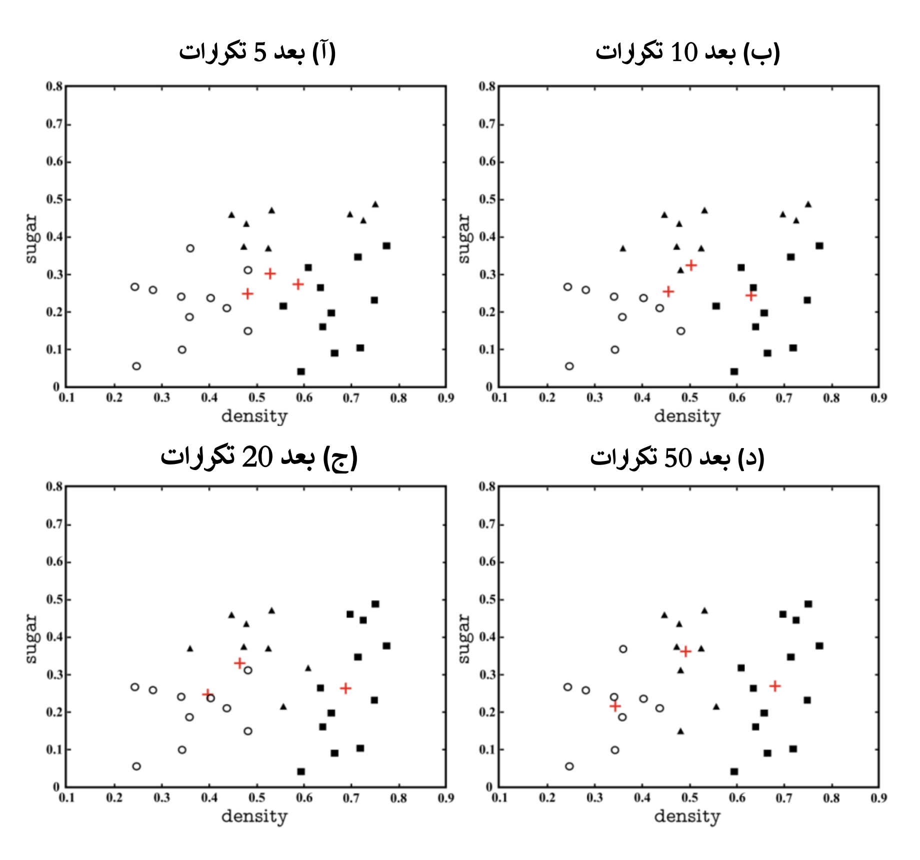
$$\acute{\mu}_1 = (0.491,0.251); \acute{\mu}_2 = (0.571,0.281); \acute{\mu}_3 = (0.534,0.295)$$

$$\acute{\Sigma}_1 = \begin{pmatrix} 0.025 & 0.004 \\ 0.004 & 0.016 \end{pmatrix}$$

$$\acute{\Sigma}_2 = \begin{pmatrix} 0.023 & 0.004 \\ 0.004 & 0.017 \end{pmatrix}$$

$$\acute{\Sigma}_3 = \begin{pmatrix} 0.024 & 0.005 \\ 0.005 & 0.016 \end{pmatrix}$$

تتكرر عملية التحديث المذكورة أعلاه حتى التقارب. . يوضح الشكل 7.3 نتائج التجميع بعد تكرارات مختلفة.



الشكل 8ـ3. نتائج خوارزمية تجميع النموذج الغاوسي المختلط بعد تكرارات مختلفة في مجموعة بيانات البطيخ (الجدول 8ـ1) مع $k = 3$. يتم عرض المتجهات المتوسطة لمكون الخليط الغاوسي بالرمز "+".

السيناريوهات المستخدمة في نموذج غاوسي المختلط

- في حالة تحليل السلاسل الزمنية ، يمكن استخدام نموذج غاوسيان المختلط لاكتشاف كيفية ارتباط التقلبات بالاتجاهات والضوضاء ، والتي يمكن أن تساعد في التنبؤ بأسعار الأسهم المستقبلية. يمكن أن تتكون مجموعة واحدة من اتجاه في سلسلة زمنية ، في حين أن مجموعة أخرى يمكن أن يكون لها ضوضاء وتقلبات من عوامل أخرى مثل الموسمية أو الأحداث الخارجية التي تؤثر على أسعار الأسهم. لفصل هذه المجموعات ، يمكن استخدام نموذج مختلط غاوسي. لأنه بدلاً من مجرد تقسيم البيانات إلى قسمين ، مثل K-means ، فإنها توفر احتمالًا لكل فئة.
- يحدث عندما تكون هناك مجموعات مختلفة في مجموعة بيانات ولا يمكن تصنيفها على أنها تنتمي إلى مجموعة أو أخرى. يمكن استخدام نموذج الخليط غاوسي في هذه الحالة. هذا لأنهم وجدوا نماذج غاوسيان التي تصف كل مجموعة بشكل أفضل وتوفر إمكانية لكل مجموعة والتي تكون مفيدة عند وضع علامات على المجموعات.
- مثال آخر يمكن أن يكون فيه النموذج المختلط الغاوسي مفيداً عندما نريد اكتشاف المجموعات الأساسية للفئات مثل أنواع السرطان أو عوامل الخطر المرتبطة بأنواع مختلفة من السرطان.

تطبيق نموذج جاوس المختلط

هناك العديد من المشكلات المختلفة في العالم الحقيقي التي يمكن حلها باستخدام نماذج غاوسي المختلطة. تعد النماذج المختلطة من غاوسيان مفيدة للغاية عندما تكون هناك مجموعة بيانات كبيرة ويصعب العثور على المجموعات. هذا هو المكان الذي يمكن فيه للنماذج المختلطة غاوسي العثور على مجموعات غاوسيان بشكل أكثر كفاءة من خوارزميات التجميع الأخرى مثل k-mean.

بعض المشاكل التي يمكن حلها باستخدام النماذج المختلطة غاوسي مذكورة أدناه:

- **البحث عن أنماط في مجموعات البيانات الطبية:** يمكن استخدام النماذج المختلطة الغاوسية لتصنيف الصور إلى فئات متعددة بناءً على محتواها أو للعثور على أنماط محددة في مجموعات البيانات الطبية.
- **نمذجة الظواهر الطبيعية:** يمكن استخدام النماذج المختلطة الغاوسية لنمذجة الظواهر الطبيعية حيث يتم تحديد أن الضوضاء تتبع التوزيعات الغاوسية.
- **تحليل سلوك العملاء:** يمكن استخدام النماذج المختلطة الغاوسية لتحليل سلوك العملاء في التسويق للتنبؤ بالمشتريات المستقبلية بناءً على البيانات السابقة.
- **التنبؤ بسعر السهم:** هناك مجال آخر تستخدم فيه النماذج المختلطة غاوسي وهو التمويل، والذي يمكن تطبيقه على السلاسل الزمنية لأسعار الأسهم. يمكن استخدام النماذج

المختلطة غاوسي لتحديد نقاط التغيير في بيانات السلاسل الزمنية وللمساعدة في العثور على معالم أسعار الأسهم أو حركات السوق الأخرى التي يصعب اكتشافها بسبب التقلبات والضوضاء.

- **تحليل بيانات التعبير الجيني:** يمكن استخدام نماذج غاوس المختلطة لتحليل بيانات التعبير الجيني. على وجه الخصوص ، يمكن استخدام النماذج المختلطة غاوسيان لتحديد الجينات المعبر عنها بين حالتين ولتحديد الجينات التي قد تلعب دورًا في نمط أو حالة مرضية معينة.

مزايا التجميع الغاوسي المختلط

- يستخدم نهجًا احتماليًا ويعرض الاحتمالات لكل نقطة من البيانات التي تنتمي إلى المجموعات.
- يمكن التعرف على المجموعات غير الكروية بأقطار مختلفة.
- انها ليست حساسة للمتغيرات على مستويات مختلفة.

إذا كنت تبحث عن طريقة فعالة للعثور على أنماط في مجموعات البيانات المعقدة أو للمساعدة في نمذجة الظواهر الطبيعية مثل الكوارث الطبيعية أو تحليل سلوك العملاء في التسويق الخاص بك ، يمكن أن تكون النماذج المختلطة غاوسيان اختيارًا جيدًا.

عيوب التجميع الغاوسي المختلط

- يتطلب مجموعات بيانات كبيرة ومن الصعب تقدير عدد المجموعات.
- نظرًا لعشوائية الغاوسيين الأوائل ، فمن المحتمل أن تتقارب مع نموذج محلي مثالي.
- إنها حساسة للحالات البعيدة.

التجميع الهرمي

في القسم السابق ، رأينا كيف يجد التجميع المستند إلى النموذج الأولي مراكز k في مساحة السمات ويقوم بتحديثها بشكل متكرر للعثور على مجموعة من العناقيد. التجميع الهرمي له نهج مختلف ، وكما يوحي اسمها ، فإن التسلسل الهرمي يطور مجموعات في شكل أشجار. يُعرف هذا الهيكل على شكل شجرة **بالرسم الشجري**[1]. بدلاً من الحصول على ناتج "سلس" من المجموعات ، فإنه يعطينا تجميعًا هرميًا لشجرة من المجموعات. نتيجة لذلك ، يوفر التجميع الهرمي نظرة ثاقبة لهياكل المجموعات المعقدة أكثر من طرق التجميع المسطحة مثل k-means.

[1] dendrogram

ومن ثم ، فإن الميزة الرئيسية للتسلسل الهرمي على النهج القائم على النموذج الأولي هي أننا نكتسب فهمًا أكثر دقة لهيكل بياناتنا ، وغالبًا ما يكون هذا النهج قادرًا على إعادة بناء التسلسلات الهرمية الحقيقية في الطبيعة. على سبيل المثال ، تخيل أننا نقوم بتسلسل الجينوم لجميع سلالات القطط. يمكننا أن نفترض بسهولة أن جينوم السلالة يشبه جينوم السلالة (السلالات) التي اشتُق منها أكثر من جينوم السلالة التي لم يُشتق منها. إذا طبقنا التجميع الهرمي على هذه البيانات ، فيمكن تفسير التسلسل الهرمي ، الذي يمكن تصوره على شكل شجرة تين ، للإشارة إلى الأجناس المشتقة من الأجناس الأخرى.

هناك طريقتان لمحاولة تعلم التسلسل الهرمي للبيانات:

- **تراكمي**[1]: نهج تصاعدي تلتقط فيه الخوارزمية جميع نقاط البيانات كمجموعات فردية وتدمجها طالما بقيت المجموعة.
- **الانقسام**[2]: يعمل هذا النهج عكس النهج التراكمي وهو من أعلى إلى أسفل. أي أنه يبدأ بكل العناصر في مجموعة ويقسمها مرة أخرى إلى مجموعات حتى يصبح كل عنصر في المجموعة الخاصة به.

الرسم الشجري Dendrogram

يمكن تمثيل المجموعات الهرمية بواسطة شجرة ثنائية متجذرة. تمثل عُقد الشجرة مجموعات أو عناقيد، وتمثل العقد الجذرية مجموعة البيانات بأكملها. أيضًا، تمثل العقد النهائية لكل منها ملاحظة واحدة (مجموعات مفردة). كل عقدة غير طرفية لها عقدتان ابنتان.

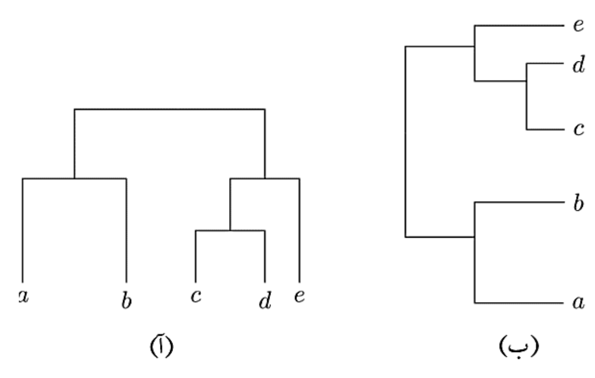
الرسم الشجري هي مخطط شجرة يستخدم لإظهار ترتيب المجموعات التي تم إنشاؤها بواسطة نظام التجميع الهرمي. يمكن تصوير الشجرة مع عقدة الجذرفي الأعلى والفروع تنمو عموديًا لأسفل (الشكل (أ) 8ـ4). قد تمتد أيضًا مع عقدة الجذر على اليسار ونمو الفروع الأفقية إلى اليمين (الشكل (ب) 8ـ4).

يوضح الشكل 8ـ5 شجرة لمجموعات البيانات $\{a, b, c, d, e\}$. لاحظ أن العقدة الجذرية تمثل مجموعة البيانات بأكملها، وأن العقد النهائية تمثل الملاحظات الفردية. ومع ذلك، يتم تقديم الرسوم الشجرية بتنسيق أبسط حيث يتم عرض العقد النهائية فقط (أي العقد التي تمثل مجموعات مفردة) بشكل صريح. يوضح الشكل 8ـ4 رسمًا مبسطًا للرسم الشجريفي الشكل 8-5.

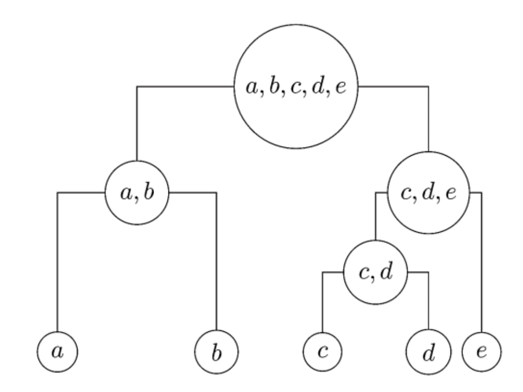
---

[1] Agglomerative

[2] Divisive



الشكل 8ـ4. طرق مختلفة لعرض الرسم الشجري



الشكل 8ـ5. الرسم الشجري لمجموعات البيانات $\{a, b, c, d, e\}$

التجميع الهرمي التراكمي

يبدأ هذا النهج من خلال النظر في كل عينة في مجموعة البيانات كمجموعة أولية. في كل جولة ، يتم بعد ذلك دمج أقرب مجموعتين في شكل مجموعة جديدة ، وتتكرر هذه العملية حتى يصل عدد المجموعات إلى قيمة محددة مسبقًا. لذلك ، يمكن تلخيص خطوات هذه الخوارزمية على النحو التالي:

1. **تحديد معيار المسافة (محدد من قبل المستخدم) بين كل مجموعة.**
2. **دمج أكثر المجموعات تشابهًا في مجموعة واحدة.**
3. **كرر الخطوتين 1 و 2 حتى تكون جميع العينات في نفس المجموعة.**

كيفية عمل هذه الخوارزمية موضحة في الشكل 8ـ6. نظرًا لوجود 9 أمثلة في هذا الشكل ، نبدأ بـ 9 مجموعات. تحسب هذه الخوارزمية معيار المسافة بين كل مجموعة وتدمج المجموعات الأكثر تشابهًا مع بعضها البعض. تستمر هذه العملية حتى تصبح جميع العينات في العنقود النهائي.

المفتاح هنا هو كيفية قياس المسافة بين المجموعات. نظرًا لأن كل مجموعة عبارة عن مجموعة من نقاط البيانات ، يجب علينا تحديد قياس المسافة للمجموعات. ومن ثم ، يتم اتخاذ قرار دمج أو عدم دمج مجموعتين عن طريق قياس الاختلاف بين المجموعات. على سبيل المثال، بالنظر إلى مجموعتي C_i و C_j، يمكننا تحديد المسافات التالية:

- اقصر مسافة:

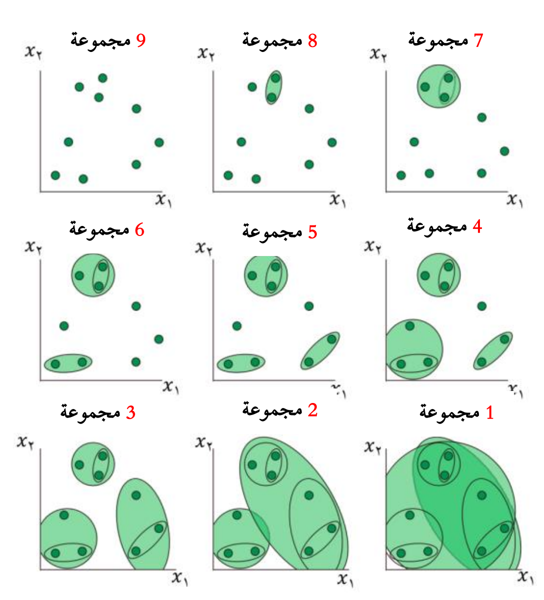
$$d_{min}(C_i, C_j) = min_{x\epsilon C_i, z\epsilon C_j} dist(x, z)$$

- اطول مسافة:

$$d_{max}(C_i, C_j) = max_{x\epsilon C_i, z\epsilon C_j} dist(x, z)$$

- متوسط المسافة:

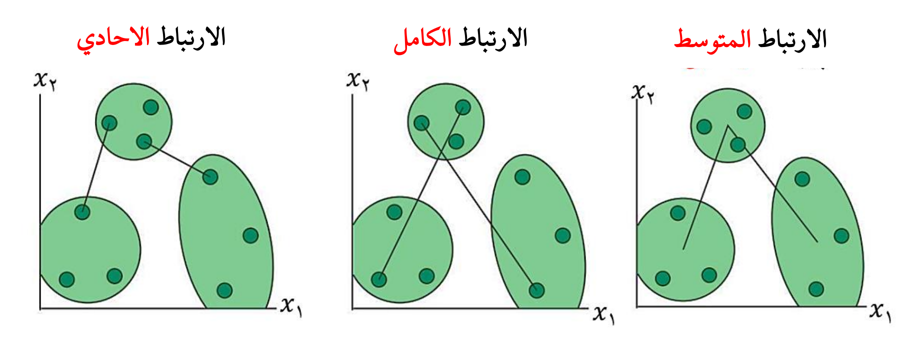
$$d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x\epsilon C_i} \sum_{z\epsilon C_j} dist(x, z)$$



الشكل 8ـ6. خوارزمية التجميع الهرمي

يتم تحديد أقصر مسافة بين مجموعتين من خلال أقرب عينتين. يتم تحديد أطول مسافة بواسطة عينتين بعيدتين عن المجموعات. يتم تحديد متوسط المسافة بواسطة جميع العيناتفي كلتا المجموعتين. عندما يتم قياس مسافات المجموعةفي d_{min}، d_{max} يا d_{avg} ، تسمى

الخوارزمية **أحادية الارتباط**[1] **أو ارتباط كامل**[2] **أو ارتباط متوسط**[3] ، على التوالي. يتم عرض كل من هذه الروابطفي الشكل 8ـ7.



الشكل 8ـ7. طرق الارتباط المختلفة. يأخذ الارتباط المسافة بين أقرب العناصر من مجموعتين كمسافة بين تلك المجموعات. يأخذ الارتباط الكامل المسافة بين الحالة الأبعد لمجموعتين كمسافة بين تلك المجموعات. يأخذ الارتباط المتوسط متوسط المسافة بين جميع حالات مجموعتين كمسافة بين تلك المجموعات.

مثال. بالنظر إلى مجموعات البيانات $\{a, b, c, d, e\}$ومصفوفة المسافة أدناه، نريد إنشاء الرسم الشجري باستخدام التجميع الهرمي كامل الارتباط:

	a	*b*	*c*	*d*	*e*
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

يستخدم التجميع الهرمي كامل الارتباط "معادلة **أطول مسافة**"، أي المعادلة التالية، لحساب المسافة بين مجموعتين C_j و C_i:

$$d_{max}(C_i, C_j) = max_{x\epsilon C_i, z\epsilon C_j} dist(x, z)$$

1. وفقًا لمجموعة البيانات $\{a, b, c, d, e\}$، فإن المجموعات الأولية تساوي:

$$C_1: \{a\}, \{b\}, \{c\}, \{d\}, \{e\}$$

---

[1] single-linkage

[2] complete-linkage

[3] average-linkage

2. يوضح الجدول أدناه المسافات بين المجموعات المختلفة في C_1:

	a	**b**	**c**	**d**	**e**
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

في الجدول أعلاه، الحد الأدنى للمسافة هو المسافة بين العناقيد $\{c\}$ و $\{e\}$. ومن ثم، فإننا ندمج و $\{c\}$ و $\{e\}$.

وفقاً لذلك، فإن المجموعة الجديدة من العناقيد تساوي:

$$C_2: \{a\}, \{b\}, \{d\}, \{c, e\}$$

3. نحسب المسافة $\{c, e\}$ من المجموعات الأخرى:

$$dist(\{c, e\}, \{a\}) = \max\{dist(c, a), dist(e, a)\} = \max\{3,11\} = 11$$
$$dist(\{c, e\}, \{b\}) = \max\{dist(c, b), dist(e, b)\} = \max\{7,10\} = 10$$
$$dist(\{c, e\}, \{d\}) = \max\{dist(c, d), dist(e, d)\} = \max\{9,8\} = 9$$

وفقاً لذلك ، يوضح الجدول أدناه المسافات بين المجموعات المختلفة في C_2.

	a	**b**	**d**	**c, e**
a	0	9	6	11
b	9	0	5	10
d	6	5	0	9
c, e	11	10	9	0

في الجدول أعلاه، الحد الأدنى للمسافة هو المسافة بين العناقيد $\{b\}$ و $\{d\}$. ومن ثم، فإننا ندمج $\{b\}$ و $\{d\}$.

وفقاً لذلك، فإن المجموعة الجديدة من العناقيد تساوي:

$$C_3: \{a\}, \{b, d\}, \{c, e\}$$

4. نحسب المسافة $\{b, d\}$ من المجموعات الأخرى:

$$dist(\{b, d\}, \{a\}) = \max\{dist(b, a), dist(d, a)\} = \max\{9,6\} = 9$$
$$dist(\{b, d\}, \{c, e\}) = \max\{dist(b, c), dist(b, e), dist(d, c), dist(d, e)\}$$
$$= \max\{7,10,9,8\} = 10$$

وفقاً لذلك ، يوضح الجدول أدناه المسافات بين المجموعات المختلفة في C_2.

	a	**b, d**	**c, e**
a	0	9	11
b, d	9	0	10
c, e	11	10	0

في الجدول أعلاه، الحد الأدنى للمسافة هو المسافة بين العناقيد $\{a\}$ و $\{b,d\}$. ومن ثم، فإننا ندمج $\{a\}$ و $\{b,d\}$.

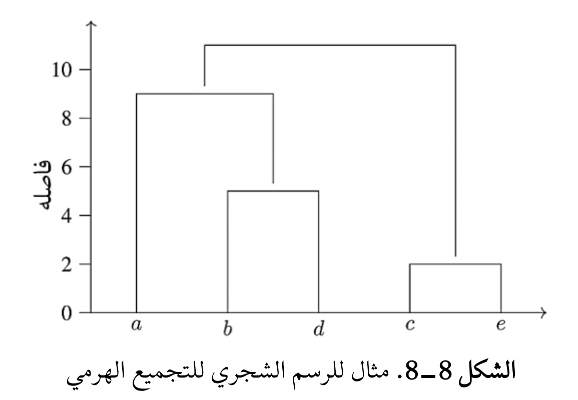
وفقًا لذلك، فإن المجموعة الجديدة من العناقيد تساوي:

$$C_4:\{a,b,d\},\{c,e\}$$

5. تبقى مجموعتان فقط. وبالتالي ، نقوم بدمجها معًا لتشكيل مجموعة واحدة تحتوي على جميع نقاط البيانات. لدينا:

$$dist(\{a,b,d\},\{c,e\})$$
$$= \max\{dist(a,c), dist(a,e), dist(b,c), dist(b,e), dist(d,c), dist(d,e)\}$$
$$= \max\{3,11,7,10,9,8\} = 11$$

6. يوضح الشكل 8ـ8 الرسم الشجري للتجميع الهرمي التراكمي.



الشكل 8ـ8. مثال للرسم الشجري للتجميع الهرمي

التجميع الهرمي التقسيمي

بخلاف التجميع التراكمي ، يبدأ التجميع التقسيمي بكل العناصر الموجودة في الكتلة ويقسمها مرة أخرى إلى مجموعات أصغر وأصغر ، حتى يصبح كل عنصر في مجموعته الخاصة. يصعب إيجاد القسمة المثلى في أي مرحلة من مراحل التجميع. ومن ثم ، فإن التجميع التقسيمي يستخدم **نهجًا مبتكرًا[1] (استكشافيًا).**

وفقًا لذلك ، في كل مرحلة من مراحل التجميع ، يتم اختيار المجموعة ذات القطر الأكبر. قطر المجموعة هي أكبر مسافة (اختلاف) بين عينتيها. ومن ثم ، فإن الخوارزمية تجد العينة في هذه المجموعة التي لديها أعلى متوسط مسافة مع العينات الأخرى في المجموعة. هذه الحالة الأكثر تباينًا تخلق مجموعتها المنقسمة الخاصة بها. تتكرر هذه العملية حتى تصبح جميع العناصر في المجموعة الخاصة بها. بشكل أساسي ، يطبق التجميع الهرمي التقسيمي تجميع k-mean(مع k = 2) في كل مستوى من التسلسل الهرمي لتقسيم كل مجموعة.

[1] heuristic approach

لا يوجد سوى تطبيق واحد لتجميع المجموعات يسمى خوارزمية DIANA. يتم استخدام التجميع التراكمي بشكل أكثر شيوعًا وهو أقل تكلفة من الناحية الحسابية من خوارزمية DIANA. ومع ذلك ، لا يمكن إصلاح الأخطاء التي حدثت في بداية التجميع الهرمي في الجزء السفلي. وبالتالي ، في حين أن التجميع التجميعي قد يكون أفضل في العثور على مجموعات صغيرة ، قد يكون DIANA أفضل في العثور على مجموعات كبيرة.

خوارزمية DIANA

الخطوات الكاملة لخوارزمية DIANA هي كما يلي:

الخطوة 1. افترض أن المجموعة C_l ستقسم إلى مجاميع (عناقيد) C_i و C_j.

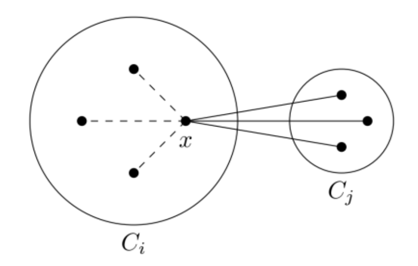
الخطوة 2. اذا كان لدينا: $C_i = C_l$ و $C_j = \emptyset$.

الخطوة 3. لكل كائن $x \in C_i$:

(آ): في التكرار الأول ، احسب متوسط المسافة x لجميع الكائنات الأخرى.

(ب): بالنسبة للتكرارات المتبقية ، قم بإجراء الحساب التالي:

$$D_x = avg\{dist(x,y): y \epsilon C_i\} - avg\{dist(x,y): y \epsilon C_j\}$$



الشكل 8ـ9. (متوسط الخط المستمر) ـ (متوسط الخط المتقطع) = D_x

الخطوة 4.

(آ): للتكرار الأول ، حرك الكائن بأقصى مسافة متوسطة إلى C_j.

(ب): بالنسبة للتكرارات المتبقية ، أوجد كائن x في C_i حيث يكون D_x هو الأكبر. إذا كانت $D_x > 0$ ، انقل x إلى C_j.

الخطوة 5. كرر الخطوتين 3ـ(ب) و 4ـ(ب) حتى تصبح جميع اختلافات D_x سالبة. ثم يتم تقسيم C_l إلى C_i و C_j.

الخطوة 6. اختر الكتلة الأصغر ذات القطر الأكبر. ثم قسّم هذه المجموعة وفقًا للخطوات من 1 إلى 5.

الخطوة 7. كرر الخطوة 6 حتى تحتوي كل المجموعات على كائن واحد فقط.

مثال. بالنظر إلى مجموعات البيانات $\{a, b, c, d, e\}$ ومصفوفة المسافة أدناه، نريد إنشاء رسم شجري مع مجموعات هرمية باستخدام خوارزمية DIANA:

	a	**b**	**c**	**d**	**e**
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

1. اولا لدينا: $C_l = \{a, b, c, d, e\}$. ايضا:
$C_i = C_l$ و $C_j = \emptyset$
2. التقسيم:

(آ): التكرار الأولي.

احسب متوسط الاختلاف بين الكائنات في C_i مع الكائنات الأخرى في C_i.

متوسط الاختلاف a:

$$= \frac{1}{4}\left(dist(a, b) + dist(a, c) + dist(a, d) + dist(a, e)\right)$$

$$= \frac{1}{4}(9 + 3 + 6 + 11) = 7.25$$

بنفس الطريقة لدينا:

متوسط الاختلاف $b = 7.75$

متوسط الاختلاف $c = 5.25$

متوسط الاختلاف $d = 7.00$

متوسط الاختلاف $e = 7.75$

أعلى متوسط مسافة هو 7.75 وهناك كائنان متماثلان. نختار واحد منهم كما نريد. نختار b وننقله إلى C_j. لدينا الآن:

$$C_i = \{a, c, d, e\}, C_j = \emptyset \cup \{b\} = \{b\}$$

(ب): التكرارات المتبقية.

- التكرار الثاني.

$$D_a = \frac{1}{3}\left(d(a, c) + d(a, d) + d(a, e)\right) - \frac{1}{1}\left(d(a, b)\right)$$

$$= \frac{20}{3} - 9 = -2.33$$

$$D_c = \frac{1}{3}\left(d(c, a) + d(c, d) + d(c, e)\right) - \frac{1}{1}\left(d(c, b)\right)$$

$$= \frac{14}{3} - 7 = -2.33$$

$$D_d = \frac{1}{3}\big(d(d,a) + d(d,c) + d(d,e)\big) - \frac{1}{1}\big(d(d,b)\big)$$
$$= \frac{23}{3} - 7 = 0.67$$
$$D_e = \frac{1}{3}\big(d(e,c) + d(e,c) + d(e,d)\big) - \frac{1}{1}\big(d(e,b)\big)$$
$$= \frac{21}{3} - 7 = 0$$

D_d هو الأكبر و $D_d > 0$. لذلك ، ننقل d الى C_j. لدينا الآن:

$C_i = \{a,c,e\}\,, C_j = \{b\} \cup \{d\} = \{b,d\}$

- التكرار الثالث.

$$D_a = \frac{1}{2}\big(d(a,c) + d(a,e)\big) - \frac{1}{2}\big(d(a,b) + d(a,d)\big)$$
$$= \frac{14}{2} - \frac{15}{2} = -0.5$$
$$D_c = \frac{1}{2}\big(d(c,a) + d(c,e)\big) - \frac{1}{2}\big(d(c,b) + d(c,d)\big)$$
$$= \frac{5}{2} - \frac{16}{2} = -13.5$$
$$D_e = \frac{1}{2}\big(d(e,c) + d(e,c)\big) - \frac{1}{2}\big(d(e,b) + d(e,d)\big)$$
$$= \frac{13}{2} - \frac{18}{2} = -2.5$$

الكل سلبي. لذلك نتوقف ونشكل عناقيد C_i و C_j.

3. لقسمة C_i و C_j، نحسب قطرهما:

$$diameter(C_i) = \max\{d(a,c), d(a,e), d(c,e)\}$$
$$= \max\{3,11,2\}$$
$$= 11$$
$$diameter(C_j) = \max\{d(b,d)\}$$
$$= 5$$

إنها مجموعة ذات قطر أكبر C_i. لذلك نقسم الآن C_i. بأخذ $C_l = \{a,c,e\}$، تُترك الحسابات المتبقية للقارئ كتمرين.

التعقيد الزماني للتجميع الهرمي

تختلف التقنيات التراكمية والتقسمية اختلافًا كبيرًا في احتياجاتها الحسابية. لنفترض أن لدينا نقاط بيانات N ، ومن ثم فإن عدد عمليات الدمج المحتملة التي يجب أن تراعيها خوارزمية

تراكمية في الخطوة الأولى $\frac{N(N-1)}{2}$ (يمكن دمج نقطتي البيانات لتشكيل مجموعة يتم تشكيلها في الحجم الثاني). يجب إجراء إجمالي عمليات الدمج $N-1$ لإنشاء تسلسل هرمي مجموعة كامل ، ويجب مراعاة إجمالي عمليات الدمج المحتملة $O(N^3)$. بشكل عام ، يتطلب النهج التراكمي $O(N^3)$ أو $O(N^2 logN)$ عند الحاجة ، اعتمادًا على ما إذا كان يجب حساب جميع المسافات بين جميع المجموعات بعد كل عملية دمج. في المقابل ، فإن عدد الأقسام المحتملة التي يجب أخذها في الاعتبار في خوارزمية التقسيم وحدها في الخطوة الأولى هو $O(2^N)$. لذلك، فإن خوارزميات التقسيم التي تأخذ في الاعتبار كل قسم ممكن للعثور على القسم الأمثل **مستعصية على الحل**[1] لمجموعات البيانات متوسطة الحجم.

مزايا و وعيوب التجميع الهرمي

المزايا

- ليست هناك حاجة لتحديد عدد المجموعات.
- يمكن أن يوفر لك الرسم الشجري معلومات مفيدة.
- من السهل فهمه وتنفيذه.

معايب

- إنه حساس للبيانات على مستويات مختلفة.
- من وجهة نظر حسابية ، فإنه يكلف الكثير في مجموعة كبيرة من البيانات.
- إنه حساس للحالات البعيدة.
- نادرا ما يعطي الحل الأفضل.
- مع مجموعات البيانات الكبيرة ، من الصعب تحديد العدد المناسب من مجموعات الرسم الشجري.

التجميع على أساس الكثافة

نهج التجميع القائم على الكثافة هو طريقة قادرة على العثور على مجموعات من الشكل المطلوب، وكما يوحي اسمها ، تستخدم كثافة العينة لتعيين عضوية المجموعة. تفترض هذه الخوارزميات أنه يمكن تحديد بنية التجميع بواسطة كثافة توزيعات العينة. عادةً ما تقوم خوارزميات التجميع المعتمدة على الكثافة بتقييم العلاقة بين العينات من منظور الكثافة وتوسيع المجموعات عن طريق إضافة عينات ذات صلة.

[1] intractable

هناك عدة طرق لقياس الكثافة ، ولكن يمكننا تعريفها على أنها عدد العينات لكل وحدة حجم لمساحة الميزة. يمكن القول أن مناطق المساحة الميزات التي تحتوي على العديد من العينات (تقع بالقرب من بعضها البعض) تتميز بكثافة عالية ، في حين أن مناطق المساحة الميزة التي تحتوي على عناصر قليلة أو لا تحتوي على عناصر ذات كثافة منخفضة. ينص حدسنا هنا على أن مجموعات الميزات يتم تمثيلها في مجموعة بيانات ذات مناطق عالية الكثافة ومفصولة بمناطق منخفضة الكثافة. تحاول خوارزميات التجميع المعتمدة على الكثافة تعلم هذه مناطق الميزة عالية الكثافة وتقسيمها إلى مجموعات. تتميز خوارزميات التجميع المعتمدة على الكثافة بالعديد من المزايا الجيدة. تتطلب الخوارزمية القائمة على الكثافة مسحًا واحدًا فقط لمجموعة البيانات الأصلية ويمكنها إدارة الضوضاء. بالإضافة إلى ذلك ، فإن عدد المجموعات غير مطلوب في هذه الطريقة ، حيث يمكن لخوارزميات التجميع المعتمدة على الكثافة أن تكتشف تلقائيًا عدد المجموعات.

على عكس العديد من خوارزميات التجميع التقليدية الأخرى ، فإن خوارزميات التجميع القائمة على الكثافة لديها القدرة على التعامل مع الحالات البعيدة. في التجميع المعتمد على الكثافة ، تُعتبر النقاط البعيدة عينات تنتمي إلى مناطق متفرقة (العزلة) وبالتالي تؤدي إلى الحدس الذي تم إنشاؤه بواسطة آليات مختلفة مقارنة بالعينات الأخرى.

تعتمد طريقة التجميع الأكثر شهرةً والأكثر شيوعًا على كثافة DBSCAN ، الموضحة في هذا القسم.

خوارزمية DBSCAN

DBSCAN هي خوارزمية تجميع تعتمد على الكثافة والتي تحدد كثافة توزيعات العينة مع زوج من معاملات "الجوار" ε, μ. لفهم خوارزمية DBSCAN ، يجب أولاً فهم هذين المعاملين. تبدأ الخوارزمية بتحديد عينة من البيانات والبحث عن عناصر أخرى داخل دائرة نصف قطرها البحث. هذا معامل لنصف قطر إبسيلون ε. يحدد المعامل μ الحد الأدنى لعدد النقاط (العناصر) التي يجب أن تمتلكها المجموعة لإنشاء مجموعة. لذلك ، يعد المعامل μ عددًا صحيحًا. إذا كانت حالة معينة بها على الأقل حالات μ داخل نصف قطر إبسيلون ، فإن هذه الحالة تعتبر نقطة مركزية.

وفقًا لمجموعة البيانات $D = \{x_1, x_2, \dots, x_m\}$ نحدد المفاهيم التالية:

- **εــ الجوار:** εــ الجوار $p \epsilon D$ ، المشار إليه بـ $N_\varepsilon(p)$ ، يعرف على النحو التالي:

$$N_\varepsilon(p) = \{q \epsilon D | dist(p, q) \leq \varepsilon\}$$

- **خصائص نقاط النواة:** يتم تصنيف كل عينة في D كنقطة محورية أو نقطة حدية أو نقطة ضوضاء ، اعتمادًا على الجوار. تعتبر العينة p نقطة مركزية إذا كانت تحتوي على أكثر من عينات μ في جوارها. إذا كانت p تحتوي على أقل من عينات μ داخل جوارها ε ولم يكن

أي من جيرانها عينات مركزية ، فسيتم تصنيف pعلى أنها ضوضاء أو عينة بعيدة. خلاف ذلك ، تسمى p عينة حدودية.

- **قابلية الوصول مباشرة إلى الكثافة:** يمكن الوصول إلى عينة $q \epsilon D$ مباشرة من العينة $p \epsilon D$ ويُشار إليها بـ $q \rhd p$ ، إذا وفقط إذا، $\mu \geq |N_\varepsilon(p)|$ و $q \epsilon N_\varepsilon(p)$.
- **متصل بالكثافة:** العيّنتان q و p متصلتان بالكثافة ويُشار إليهما بـ $q \bowtie p$ ، إذا كان هناك تسلسل $x_1, x_2, \dots, x_m$ للعينات بحيث:

$$p \lhd x_1 \lhd \cdots \rhd x_m \rhd q \text{ و } \forall_{x_i}: |N_\varepsilon(p)| \geq \mu.$$

- **المجموعة:** يتم تعريف الكتلة على أنها أقصى مجموعة من العينات المرتبطة بالكثافة وتتكون من عينات مركزية وعينات حدودية. في DBSCAN ، يمكن أن تنتمي العينة الحدودية إلى عدة مجموعات حسب ترتيب العينات. لا تنتمي عينة الضوضاء إلى أي مجموعة وتسمى بعيدة. تسمى المجموعة الفرعية $C \subseteq D$ بالمجموعة (العنقود) إذا كان لديها الشرطان التاليان:

1. **الحد الاقصى:** $\forall_p \epsilon C: p, \forall_q \epsilon C: p \backslash C: \neg p \bowtie q$
2. **المحاذاة:** $\forall_{p,q} \epsilon C: p \bowtie q$

يستخدم DBSCAN بُنية بيانات تسمى قائمة البذور S، والتي تتضمن مجموعة من عينات الحبوب لتوسيع المجموعة. لبناء مجموعة ، يختار DBSCAN بشكل عشوائي عينة غير معالجة ويضعها في قائمة S الفارغة كتهيئة. ثم يقوم باستخراج عينة p من S بشكل مستمر والاستعلام عن النطاق في p للعثور على عينات يمكن الوصول إليها مباشرة من p ، وإذا لم تتم معالجتها بعد ، فيضعها في S. عندما تكون قائمة البذور S فارغة ، يكتمل التجميع ويبدأ التجميع الجديد. تتكرر عملية التوسع بأكملها حتى يتم تصنيف جميع العينات.

خذ مجموعة بيانات البطيخ في الجدول 8ـ1 كمثال لتوفير رؤية أكثر دقة. افترض أن معاملات الجوار هي $\varepsilon = 0.11, \mu = 5$. نبدأ بإيجاد الجوار لكل عينة حتى نتمكن من تحديد مجموعة العينات الأساسية (المركزية):

$$\Omega = \{x_3, x_5, x_6, x_8, x_9, x_{13}, x_{14}, x_{18}, x_{19}, x_{24}, x_{25}, x_{28}, x_{29}\}$$

بعد ذلك ، نختار عينة أساسية بشكل عشوائي كبذرة ونوسعها لتشمل جميع العينات المتوفرة بكثافة. هذه العينات تشكل مجموعة. بدون فقدان المجموع ، افترض أن العينة المركزية x_8 قد تم اختيارها كبذرة ، ومن ثم يتم إنتاج المجموعة الأولى على النحو التالي:

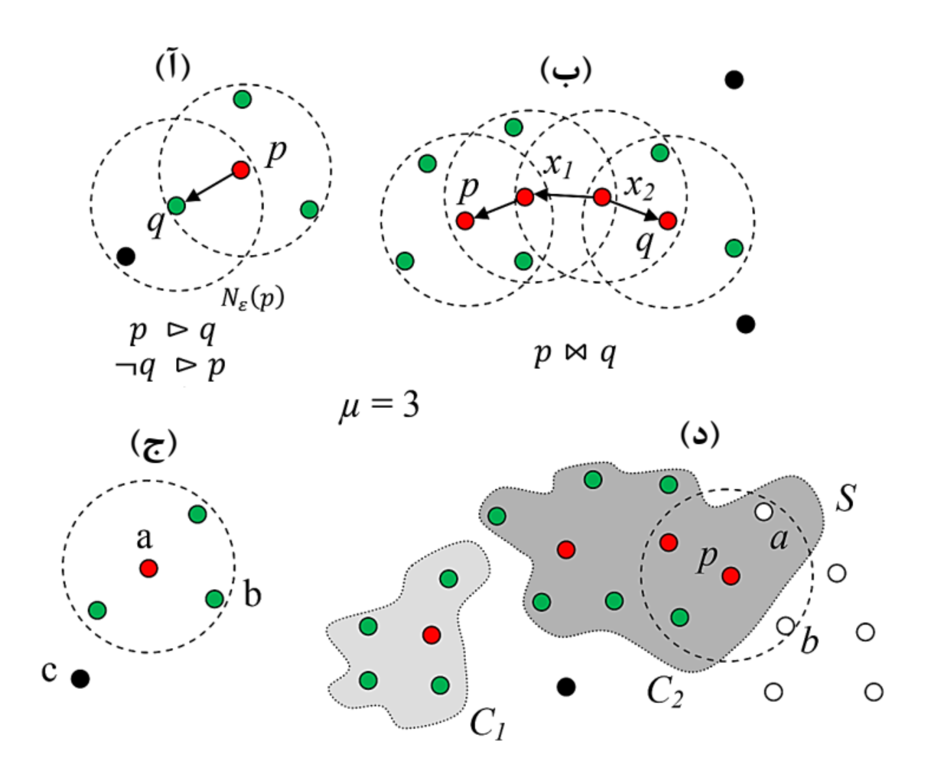
$$C_1 = \{x_6, x_7, x_8, x_{10}, x_{12}, x_{18}, x_{19}, x_{20}, x_{23}\}$$

يقوم DBSCAN بعد ذلك بإزالة كافة العينات الأصلية في C_1 من Ω :

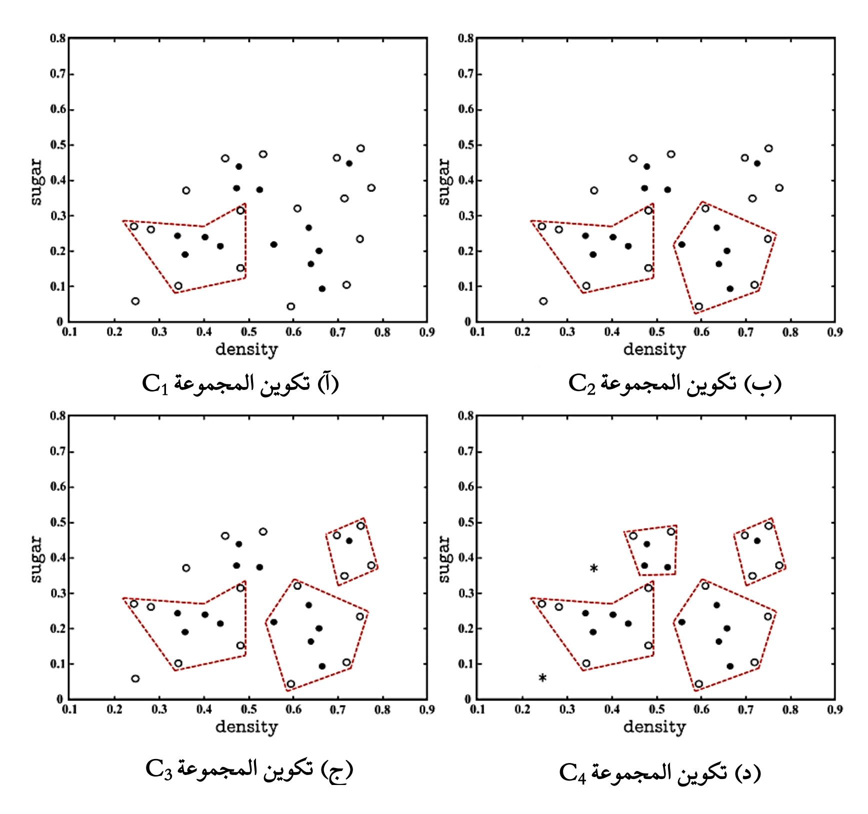
$$\Omega = \Omega \backslash C_1 = \{x_3, x_5, x_9, x_{13}, x_{14}, x_{24}, x_{25}, x_{28}\}$$

يتم بعد ذلك إنتاج المجموعة التالية عن طريق الاختيار العشوائي لعينة أساسية أخرى من Ω المحدثة كبذور. تتكرر هذه العملية حتى لا يكون هناك عنصر آخر في Ω. يوضح الشكل 8ـ11 المجموعات التي تم إنتاجها في دورات مختلفة. بالإضافة إلى C_1 ، تم إنشاء ثلاث مجموعات أخرى:

$$C_2 = \{x_3, x_4, x_5, x_9, x_{13}, x_{14}, x_{16}, x_{17}, x_{21}\}$$
$$C_3 = \{x_1, x_2, x_{22}, x_{26}, x_{29}\}$$
$$C_4 = \{x_{24}, x_{25}, x_{27}, x_{28}, x_{30}\}.$$



الشكل 8ـ10. مفاهيم DBSCAN. **(آ)** q يمكن الوصول إليها مباشرة من p . (ب) p و q مرتبطان بالكثافة. **(ج)** المثال a (أحمر) عينة مركزية (أساسية) ، b (أخضر) عينة حدودية ، c (أسود) عينة ضوضاء. **(د)** قائمة الحبوب S لتوسيع المجموعة. تقوم DBSCAN حاليًا ببناء مجموعة C_2. يتم استخراج العينة p من S وفحصها. لا تتم معالجة العينات a و b الموجودة في المنطقة المجاورة لـp وبالتالي فهي موجودة في S.



الشكل 8ـ11. نتائج الخوارزمية DBSCAN مع $\mu = 5$ و $\varepsilon = 0.11$. تمثل الرموز "●"، "○"، "*" عينات مركزية (أساسية) وعينات غير مركزية وعينات صاخبة، على التوالي. تشير الخطوط المتقطعة إلى المجموعات.

مزايا DBSCAN

- لديها مفهوم يسمى الضوضاء ، لذلك فهو أقوى في الحالات البعيدة.
- ليست هناك حاجة لتحديد عدد المجموعات مقدماً.

عيوب DBSCAN

- لا يمكن تجميع مجموعات البيانات ذات الاختلافات الكبيرة في الكثافة جيداً.
- ليس أكيداً تماماً. لذلك ، يمكن أن تكون النقاط الحدودية التي يمكن الوصول إليها من أكثر من مجموعة واحدة جزءاً من كل مجموعة.

التعقيد الزماني والمكاني لخوارزمية DBSCAN

التعقيد الزماني لخوارزمية DBSCAN يساوي (m × الوقت اللازم للعثور على نقاط في ε-الجوار) O عندما يكون m هو عدد النقاط. في أسوأ الحالات ، يكون هذا التعقيد هو $O(m^2)$. ومع ذلك ، في المساحات منخفضة الأبعاد (خاصة المساحات ثنائية الأبعاد) ، تسمح هياكل

البيانات مثل أشجار kd بالاسترداد الفعال لجميع النقاط على مسافة معينة من نقطة معينة ، ويمكن تقليل التعقيد الزمني إلى $O(mlogm)$ في المتوسط. المساحة المطلوبة لـ DBSCAN، حتى بالنسبة للبيانات عالية الأبعاد ، هي $O(m)$ ، لأنه لا يلزم تخزين سوى كمية صغيرة من البيانات لكل نقطة ، أي تسمية المجموعة وتحديد كل نقطة كمركز أو حد أو نقطة الضوضاء. من المهم أن نلاحظ أن التعقيد الزمني لمعايير التشابه بين الكائنات لم يتم النظر فيه هنا. بافتراض أن معيار تشابه الكائن له تعقيد زمني Ψ ، فإن التعقيد النهائي لـ DBSCAN هو $O(\Psi m^2)$ أو $O(\Psi mlogm)$.

التجميع مع بايتون

تجميع k-means

استيراد المكتبات

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
```

تحضير البيانات

الخطوة التالية هي تحضير البيانات التي نريد تجميعها. لنقم بإنشاء مصفوفة numpy مكونة من 20 صفًا وعمودين.

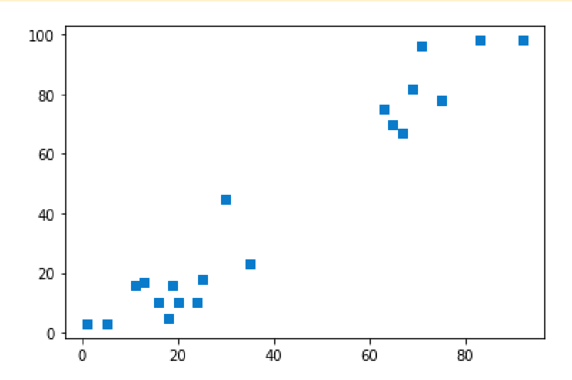
In [1]:
```
X = np.array([[1,3],
     [11,16],
     [16,10],
     [20,10],
     [35,23],
     [75,78],
     [69,82],
     [63,75],
     [65,70],
     [83,98],
     [71,96],
     [25,18],
     [18,5],
     [92,98],
     [67,67],
     [5,3],
     [13,17],
     [19,16],
     [24,10],
```

```
           [30,45]])
```

التمثيل المرئي للبيانات

لنرسم هذه النقاط ونرى ما إذا كان بإمكاننا رؤية مجموعة. للقيام بذلك ، قم بتشغيل التعليمات البرمجية التالية:

```
In [2]:  plt.scatter(X[:,0],X[:,1], label='True Position',marker = 's')
```



كما هو موضح في الشكل أعلاه ، إذا أردنا تكوين مجموعتين من نقاط البيانات ، فسنقوم على الأرجح بإنشاء مجموعة واحدة من ثماني نقاط في أعلى اليمين ومجموعة واحدة من اثنتي عشرة نقطة في أسفل اليسار. دعونا نرى ما إذا كانت خوارزمية التجميع K-mean الخاصة بنا تفعل الشيء نفسه.

إنشاء المجموعات

لتشغيل خوارزمية التجميع K-mean مع مجموعتين ، ما عليك سوى تشغيل الكود التالي:

```
In [4]:  kmeans = KMeans(n_clusters=2)
         kmeans.fit(X)
```

في السطر الأول ، تقوم بإنشاء كائن KMeans وتمرير الرقم 2 كقيمة معامل لعدد المجموعات n_clusters. بعد ذلك ، تحتاج ببساطة إلى استدعاء التابع fit على kmeans وإرسال البيانات التي تريد تجميعها. في هذا المثال ، توجد البيانات في المصفوفة X التي أنشأناها سابقًا.

الآن دعونا نرى ما هي قيم المركز التي أنشأتها الخوارزمية للمجموعات النهائية.

```
In [3]:  print(kmeans.cluster_centers_)

Out [7]: [[18.08333333 14.66666667]
          [73.125      83.        ]]
```

ناتج المصفوفة ثنائية الأبعاد هو 2 × 2. هنا يحتوي الصف الأول على قيم إحداثيات المركز الأول (14.16666667 ، 18.08333333) ويحتوي الصف الثاني على قيم إحداثيات المركز الآخر (83.0 ، 73.125).

لمشاهدة تسميات نقاط البيانات ، قم بتشغيل التعليمات البرمجية التالية:

```
In  [3]:   print(kmeans.labels_)

Out [7]:   [0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0]
```

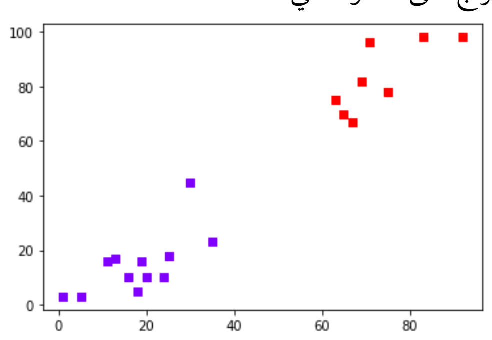
يتوافق إخراج مصفوفة أحادية البعد تتكون من 20 عنصرًا مع المجموعات المخصصة لنقاط البيانات العشرين لدينا. يتم استخدام 0 و 1 هنا فقط لتمثيل معرفات الكتلة وليس لها أهمية رياضية. إذا كانت هناك ثلاث مجموعات ، فسيتم الإشارة إلى المجموعة الثالثة بالرقم 2.

التمثيل المرئي للبيانات

دعنا نعيد نمثل نقاط البيانات على الرسم البياني ونوضح كيفية تجميع البيانات. هذه المرة نرسم البيانات جنبًا إلى جنب مع التسمية المخصصة حتى نتمكن من التمييز بين المجموعات. قم بتشغيل التعليمات البرمجية التالية:

```
In  [2]:   plt.scatter(X[:,0],X[:,1], c=kmeans.labels_, cmap='rainbow',marker = 's')
```

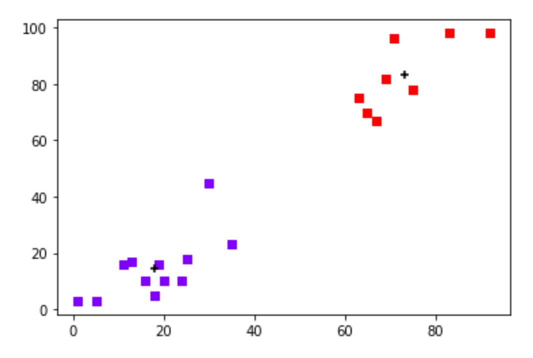
هنا نرسم العمود الأول من المصفوفة X أمام العمود الثاني، وفي هذه الحالة نرسل أيضًا kmeans.labels_ كقيمة للمعامل c الذي يتوافق مع العلامات. يتم استخدام المعامل cmap='rainbow' لتحديد نوع اللون لنقاط البيانات المختلفة ويتم إرسال المعامل marker = 's' لتمييز النقاط. الإخراج على النحو التالي:



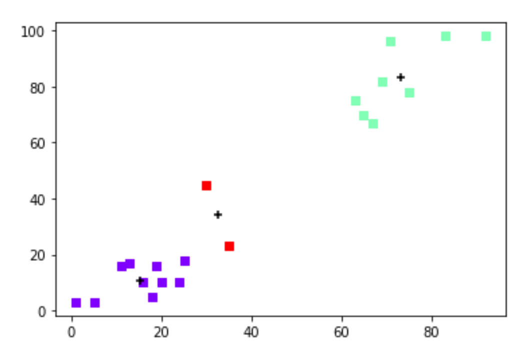
كما هو متوقع وكما هو موضح في الشكل أعلاه ، تم تجميع النقاط الاثنتي عشرة الموجودة في أسفل اليسار معًا والنقاط المتبقية في أعلى اليمين مجمعة معًا في مجموعة.

الآن دعنا نرسم النقاط على طول الإحداثيات المركزية لكل مجموعة لنرى كيف يؤثر موضع المركز على التجميع. قم بتشغيل التعليمات البرمجية التالية لرسم المخطط:

```
In  [2]:   plt.scatter(X[:,0], X[:,1], c=kmeans.labels_, cmap='rainbow',marker = 's')
           plt.scatter(kmeans.cluster_centers_[:,0] ,kmeans.cluster_centers_[:,1],
           color='black',marker = '+')
```



مرة أخرى ، نقوم بتشغيل خوارزمية K-mean مع 3 مجموعات. يتم الحصول على مخطط الإخراج التالي:



سترى ذلك مرة أخرى ، يتم وضع النقاط القريبة بجانب بعضها البعض.

كتمرين ، يمكن تجميع هذه البيانات مع خوارزميات أخرى باستخدام مكتبة scikit-Learn. يمكنك استخدام مقتطفات التعليمات البرمجية التالية. يمكنك أيضًا تغيير المعاملات المختلفة لكل خوارزمية تجميع وعرض النتائج.

التجميع الهرمي

```
In [4]: from sklearn.cluster import AgglomerativeClustering
        cluster = AgglomerativeClustering(n_clusters=2,
        affinity='euclidean', linkage='single')
        cluster.fit_predict(X)
```

التجميع DBSCAN

```
In [4]: from sklearn.cluster import DBSCAN
        clustering = DBSCAN(eps=3, min_samples=2).fit(X)
        clustering.labels_
```

التجميع الغاوسي المختلط

```
In [4]: from sklearn import mixture
        model = mixture.GaussianMixture(n_components=2,
        covariance_type='full').fit(X)
        labels = model.predict(X)
```

متى نستخدم التجميع؟

بغض النظر عن نوع البحث الذي تقوم به أو ما يقوم به نموذج التعلم الآلي الخاص بك ، سوف تحتاج إلى تقنيات التجميع في نقطة واحدة على الخط. لأنه في كثير من الحالات (في البداية على الأقل) ، فإنك تعمل مع مجموعات البيانات التي تكون في الغالب غير منظمة وغير مصنفة، ومن الضروري تجميع البيانات وإعدادها مع بعضها البعض. والأهم من ذلك ، أن التجميع هو طريقة سهلة لإجراء العديد من التحليلات السطحية التي يمكن أن تمنحك ربحًا سريعًا في مجموعة متنوعة من المجالات. على سبيل المثال ، يمكن لشركات التأمين تحديد عوامل الخطر والمواقع بسرعة وإنشاء ملف تعريف مخاطر أولي لمقدمي الطلبات.

في علم البيانات ، يمكننا استخدام تحليل المجاميع (العناقيد) لاكتساب رؤى قيمة من بياناتنا من خلال رؤية المجموعات التي تقع فيها نقاط البيانات عند تطبيق خوارزمية التجميع. بمعنى آخر ، يمكن أن يحدث التجميع وتحليلات البيانات ثورة في عملية إدارة البيانات الخاصة بك. نظرًا لأن التجميع يتم بواسطة خوارزمية ، فمن المحتمل أنك ستكون قادرًا على اكتشاف الارتباطات غير المعروفة سابقًا في البيانات التي يمكن أن تساعدك في مواجهة تحدي الأعمال من منظور جديد.

عندما يتعلق الأمر بالتنقيب في البيانات أو استخراجها ، يمكنك استخدام تجميع البيانات كأداة قائمة بذاتها لاكتساب رؤى حول توزيع البيانات أو لتعزيز المجموعات المحددة التي ترغب في مزيد من التحليل. يمكنك أيضًا استخدامه في ذكاء الأعمال لتنظيم العملاء الجدد وتنظيم المشاريع المعلقة والعديد من التطبيقات الأخرى. يساعد التجميع في جعل استخراج البيانات أكثر كفاءة عن طريق تقليل عدد عمليات المسح المطلوبة للبحث عن البيانات.

بالنسبة إلى كل العمل الرائع الذي يمكن أن يقوم به تحليل المجموعة لمشروعك ومؤسستك، هناك العديد من الأشياء التي تجعله غير مرغوب فيه عندما تبحث عن رؤى متعمقة. بعبارة أخرى، يمثل التجميع بحد ذاته تحديات كبيرة ولا يجعله مثاليًا لمهام التعلم الآلي والتحليل الأكثر تعقيداً. تكمن المشكلة الأكبر في معظم طرق التجميع في أنه على الرغم من تقسيم بياناتك مبدئيًا إلى مجموعات فرعية ، فإن الاستراتيجيات المستخدمة في بعض الأحيان لا ترتبط بالضرورة بالبيانات نفسها ، ولكن بموضعها بالنسبة إلى نقاط أخرى. على سبيل المثال ، يمكن أن يكون لتجميع K-means نتائج مختلفة جداً اعتمادًا على عدد المجموعات التي تحددها ، وعادةً لا تعمل بشكل جيد عند استخدامها مع مجموعات غير كروية. بالإضافة إلى ذلك ، فإن حقيقة

تعديل مركز الجرائم بشكل عشوائي تؤثر أيضًا على النتائج ويمكن أن تؤدي إلى مشاكل. يمكن أن تحل الخوارزميات الأخرى هذه المشكلة ، ولكن ليس بدون تكلفة.

ميل التجميع الهرمي إلى إنتاج نتائج أكثر دقة ، ولكنها تتطلب قوة حسابية كبيرة وليست مثالية عند العمل مع مجموعات البيانات الأكبر. هذه الطريقة حساسة أيضًا للقيم البعيدة ونتيجة لذلك يمكن أن تنشئ مجموعات غير صحيحة. ومع ذلك ، هذا لا يعني أنه لا يجب عليك استخدام التجميع مطلقًا ، ولكن يجب عليك استخدامه أينما ومتى يمنحك أكبر قدر من التأثير والبصيرة. أيضًا ، هناك العديد من المواقف التي لا يمنحك فيها التجميع نقطة بداية جيدة فحسب ، بل يوضح أيضًا الميزات المهمة لبياناتك التي يمكن تحسينها عن طريق التحليل المتعمق. هذه ليست سوى بعض البرامج التجريبية لتحديد الأهداف التي يمكنك استخدامها:

- **عندما تبدأ بمجموعة بيانات كبيرة غير منظمة:** مثل أدوات التعلم الأخرى غير الخاضعة للإشراف ، يمكن للتجميع التقاط مجموعات بيانات كبيرة وتنظيمها سريعًا في عناصر أكثر قابلية للاستخدام بدون تدريب. ميزة هذا هو أنه إذا كنت لا تبحث عن تحليل شامل ، يمكن أن يوفر لك التجميع إجابات سريعة حول بياناتك.
- **عندما لا تعرف عدد أو عدد الفئات التي تقع فيها بياناتك:** حتى إذا بدأت بمجموعة بيانات أكثر تنظيماً ، فقد لا تحتوي بعد على الفئة التي تريدها. يعد التجميع خطوة أولى رائعة في إعداد بياناتك ، حيث يبدأ في الإجابة عن الأسئلة الرئيسية حول مجموعة البيانات الخاصة بك. على سبيل المثال ، قد تجد أن ما كنت تعتقد أنه مجموعتان فرعيتان رئيسيتان هو في الواقع أكثر من ذلك.
- **عند تقسيم البيانات يدويًا والتعليق عليها يستغرق الكثير من الوقت والمال:** بالنسبة لمجموعة بيانات أصغر ، يكون التعليق التوضيحي والتنظيم يدويًا ممكنًا ، إن لم يكن مثاليًا. ومع ذلك ، مع بدء زيادة بياناتك ، يصبح من الصعب بشكل متزايد إضافة تعليقات توضيحية إليها. يمكن أن يؤدي التجميع (اعتمادًا على الخوارزمية التي تستخدمها) إلى تقليل وقت التعليقات التوضيحية. لأنه أقل اهتمامًا بنتائج محددة وأكثر اهتمامًا بالتصنيف الذاتي.
- **عند البحث عن الحالات الشاذة في بياناتك:** الشيء المثير للاهتمام هو أن أحد التطبيقات الأكثر قيمة للتجميع هو أنه نظرًا لأن الخوارزميات حساسة جدًا للنقاط البعيدة ، يمكن أن تعمل كمعرف للشذوذ في البيانات. يمكن أن يساعدك فهم البيانات غير الطبيعية في جعل بياناتك الحالية أكثر تناسقًا وتحقيق نتائج أكثر دقة.

ستحصل على أقصى استفادة من التجميع عندما تستخدم التجميع ليس كنموذج قائم بذاته ، ولكن كجزء من إستراتيجية أوسع لاكتشاف البيانات.

تقليل الابعاد

في التعلم الآلي، يشير "البعد" إلى عدد السمات (متغيرات الإدخال)في مجموعة البيانات. عندما يكون عدد السمات أكبر من عدد العيناتفي مجموعة البيانات الخاصة بك، يجب زيادة عدد العينات للنموذج لتحقيق أفضل أداء. بمعنى آخر، مع زيادة عدد الميزات، يزداد عدد العينات أيضًا. لأنه كلما زاد عدد الميزات التي لدينا، سنحتاج إلى المزيد من العينات حتى يتم تمثيل جميع مجموعات قيم الميزات بشكل جيدفي مجموعة البيانات. بالإضافة إلى ذلك،كلما زاد عدد الميزات، زادت إمكانية التخصيص. يعتمد نموذج التعلم الآلي الذي يتم تدريبه على عدد كبير من الميزات بشكل متزايد على البيانات التي يتم تدريبها، مما يؤدي بدوره إلى الضبط الزائد، مما يؤدي إلى ضعف الأداء على البيانات غير المرئية. يعد تجنب الضبط الزائد هو الدافع الرئيسي **لتقليل الأبعاد**[1]. كلما قل عدد الميزات التي تتمتع بها بيانات التدريب الخاصة بنا، كلما كانت افتراضات نموذجنا أقل وأبسط. لكن هذا ليس كل شيء، وهناك فوائد أكثر بكثير لتقليل الابعاد. ومع ذلك، عندما نقوم بتقليل ابعاد مجموعة البيانات، فإننا نفقد نسبة مئوية من مرونة البيانات الأصلية. ومع ذلك، لا تقلق بشأن فقدان هذه النسبة المئوية من المرونةفي البيانات الأصلية، حيث أن تقليل الابعاد له فوائد عديدة.

البيانات هى أساس أى خوارزمية للتعلم الآلى، وبدونها لا يمكن أن يحدث علم البيانات. فى بعض الأحيان، قد تحتوى مجموعة البيانات هذه على العديد من الميزات، بعضها غير مطلوب حتى. مثل هذه المعلومات الإضافية تعقد النمذجة. بالإضافة إلى ذلك، يصعب تفسير البيانات وفهمها من خلال التمثيل المرئى للبيانات نظرًا لحجمها الكبير. هذا هو المكان الذى يتم فيه تقليل الابعاد.

تعني الأبعاد الأقلفي البيانات وقت تدريب أقل وموارد حسابية أقل. تجعل مشاكل التعلم الآلي التي تنطوي على العديد من الميزات التعلم بطيئًا للغاية.في مجموعة البيانات الكبيرة، من المحتمل أن تكون معظم نقاط البيانات متباعدة. لذلك، لا يمكن تدريب الخوارزميات بفعالية وكفاءة على البيانات عالية الأبعاد (مشكلة لعنة الأبعاد).

تقليل الأبعاد يمنع مشكلة الضبط الزائد. عندما يكون هناك العديد من الميزاتفي البيانات، تصبح النماذج أكثر تعقيداً وتميل إلى الضبط الزائدفي بيانات التدريب.

تقليل الأبعاد مفيدة جداً للتمثيل المرئي للبيانات. عندما نقوم بتقليل أبعاد البيانات عالية الأبعاد إلى مكونين أو ثلاث مكونات، يمكن بسهولة رسم البياناتفي مخطط ثنائي الأبعاد أو ثلاثي الأبعاد.

---

[1] Dimensional reduction

يزيل تقليل أبعاد الضوضاءفي البيانات. من خلال الحفاظ على أهم الميزات والقضاء على الميزات الإضافية، فإنه يلغي تقليل أبعاد الضوضاءفي البيانات. نتيجة لذلك، فإنه يحسن دقة النموذج.

تعريف تقليل الابعاد

يشيرتقليل الابعاد إلى عملية تقليل عدد السمات فى مجموعة البيانات، مع الاحتفاظ بالتغييرات على مجموعة البيانات الأصلية قدر الإمكان. تعمل عملية الأبعاد بشكل أساسى على تحويل البيانات من مساحة ميزة عالية الأبعاد إلى مساحة ميزة ذات أبعاد أصغر. فى الوقت نفسه، من المهم ألا تضيع الميزات المفيدة للبيانات أثناء التحويل.

يعد تقليل الابعاد خطوة معالجة مسبقة. هذا يعنى أنه قبل تدريب النموذج، نقوم بتقليل الأبعاد.

بشكل عام، هناك طريقتان لتقليل الأبعاد: **اختيار الميزة واستخراج الميزة (التحويل)**. يحاول نهج اختيار الميزة تحديد مجموعة فرعية من الميزات المهمة وإزالة الميزات غير المهمة لتقليل تعقيد النموذج وزيادة الكفاءة الحسابية للنموذج وتقليل خطأ التعميم بسبب توليد الضوضاء.في المقابل، يحاول استخراج الميزة، المعروف أيضًا باسم تحويل الميزة، إنشاء فضاء فرعي جديد للميزة. الفكرة الرئيسية وراء الاستخراج هي ضغط البيانات بهدف الحفاظ على المزيد من المعلومات ذات الصلة.

تعريف اختيار الميزة

اختيار الميزة هو عملية التحديد التلقائى أو اليدوى لمجموعة فرعية من الميزات الأكثر ملاءمة وذات الصلة لاستخدامها فى بناء النموذج.

يتم اختيار الميزة عن طريق إضافة سمات مهمة أو حذف السمات غير ذات الصلة فى مجموعة البيانات دون تغييرها.

تعريف استخراج الميزة

استخراج الميزات هو عملية تقليل عدد الميزات فى مجموعة البيانات عن طريق إنشاء ميزات جديدة.

الأهداف الرئيسية لتقليل الأبعاد هى: تحسين دقة أداء النموذج التنبئى، لتقليل وقت الحساب وتحسين قابلية تفسير النموذج.

اختيار الميزة مقابل استخراج الميزة

تقلل كلتا الطريقتين من عدد الأبعاد، ولكن بطرق مختلفة. من المهم جداً التمييز بين هذين النوعين من الطرق. الغرض من استخراج الميزات هو تقليل عدد الميزاتفي مجموعة البيانات عن طريق إنشاء ميزات جديدة من الميزات الموجودة (ثم تجاهل الميزات الأصلية). يجب أن تكون مجموعة الميزات الجديدة المصغرة هذه قادرة على تلخيص معظم المعلومات الموجودة في مجموعة الميزات الرئيسية. بهذه الطريقة، يمكن إنشاء نسخة موجزة من الميزات الرئيسية من المجموعة الأصلية. على عكس هدف اختيار الميزات، فإنه يحتفظ بأهم الميزاتفي مجموعة البيانات ويزيل الميزات الإضافية. يتمثل الاختلاف بين اختيار الميزة واستخراج الميزةفي أن الغرض من اختيار الميزة هو ترتيب أهمية الميزاتفي مجموعة البيانات واستبعاد الميزات الأقل أهمية. بمعنى آخر، لا يتم إنشاء ميزات جديدة بهذه الطريقة.في المقابل، يؤدي استخدام استخراج الميزات إلى إنشاء مجموعة جديدة تمامًا من الميزات. يمكن تصنيف نهج استخراج الميزات إلى طرق خطية وأساليب غير خطية. تُعرف الأساليب غير الخطية أيضًا باسم التعلم المتنوع.

تقليل الأبعاد هو عملية تقليل عدد الأبعاد فى البيانات إما عن طريق إزالة الميزات الأقل فائدة (اختيار الميزة) أو تحويل البيانات إلى أبعاد أقل (استخراج الميزة).

تقنيات اختيار الميزة

يمكن اختيار الميزة يدويًا أو باستخدام تقنيات مشتركة (آلية) تم تطويرها لهذا الغرض. على سبيل المثال، افترض أنك تحاول بناء نموذج يتنبأ بوزن الأشخاص وقد جمع مجموعة كبيرة من البيانات التي تصف كل شخص. إذاكان لديك عمود يصف لون ملابس كل شخص، فهل يساعد توقع وزنه كثيرًا؟ أعتقد أننا يمكن أن نتفق بأمان على أن الأمر لن يكون كذلك. هذه ميزة يمكننا وضعها جانبًا دون أي مشاكل. عندما تكون بعض الميزات ذات صلة أو غير مرتبطة، يمكننا تحديد هذه الميزات يدويًا وتقليل الأبعاد، وعندما لا يتم تحديد هذه الميزات بوضوح، هناك العديد من التقنيات والأدوات التي يمكننا استخدامها للمساعدةفي تحديد الميزات.

يمكن أن تكون تقنيات اختيار الميزات غير خاضعة للإشراف أو خاضعة للإشراف (مثل الخوارزميات الجينية). يمكن أيضًا الجمع بين عدة طرق إذا لزم الأمر. يمكن وصف عملية اختيار الميزةفي خطوتين:

- يجمع بين أسلوب البحث لاقتراح مجموعة ميزة فرعية جديدة.
- معيار تقييم يقوم بتعيين درجة لمجموعات فرعية مختلفة.

الخوارزمية الجينية

من وجهة نظر رياضية، تتم صياغة اختيار الميزة كمشكلة تحسين مركبة. الدالة الهدف هي دالة التعميم للنموذج التنبئي، والتي يتم تمثيلها ببيان الخطأفي الميزات المحددة لمجموعة البيانات.

يقوم التحديد الشامل للسمات بتقييم 2^N مجموعات مختلفة، حيث N هو عدد السمات. تتطلب هذه العملية الكثير من العمل الحسابي وإذا كان عدد الميزات كبيرًا، فإنها تصبح غير عملية. بمعنى آخر، اختيار الميزة هو مشكلة $NP - Hard$. وبالتالي، نحتاج إلى طرق ذكية تمكن من اختيار الميزاتفي الممارسة. تعد الخوارزمية الجينية من أكثر الخوارزميات تقدمًا لاختيار الميزات.

الخوارزمية الجينية هي طريقة عشوائية لتحسين الدالة بناءً على الميكانيكا الجينية الطبيعية والتطور البيولوجي.في الطبيعة، تتطور جينات الكائنات الحية عبر الأجيال المتعاقبة لتتكيف بشكل أفضل مع البيئة. الخوارزمية الجينية هي طريقة تحسين استكشافية مستوحاة من إجراءات التطور الطبيعي. تعمل الخوارزميات الجينية على مجموعة من الأفراد لإنتاج تقديرات تقريبية أفضل وأفضل. تخلق هذه الخوارزمية مجموعة سكانية جديدةفي كل جيل عن طريق اختيار الأفراد. ثم يتم الجمع بين هؤلاء الأفراد باستخدام عوامل مستعارة من علم الوراثة الطبيعية. يمكن أيضًا تحور الأطفال. تؤدي هذه العملية إلى تطور مجتمعات أكثر ملاءمة لبيئتها من الأشخاص الذين أنشأوها.

في التعلم الآلي، للخوارزميات الجينية تطبيقان رئيسيان. الأول هو التحسين، مثل إيجاد أفضل وزن للشبكة العصبية. الحالة الثانية هي تحديد الميزة يتم الاشراف عليها. في هذه الحالة، تمثل "الجينات" الميزات الفردية، ويمثل "الكائن الحي" مجموعة من الميزات. يتم تصنيف كل كائن حيفي "السكان" على أساس درجة الملائمة[1]. تعيش الكائنات الحية الأكثر ملاءمة وتتكاثر حتى تتقارب الأجيال القليلة القادمة من السكان لحل واحد.

الميزة:

- يمكن للخوارزميات الجينية تحديد السمات بفعالية من مجموعات بيانات كبيرة جدًا ، حيث يكون البحث الشامل مستحيلًا. من المحتمل أن تكون الخوارزمية الجينية هي الخيار الأفضل بالنسبة لك عندما تحتاج إلى معالجة البيانات لخوارزمية لا تحتوي على تحديد سمة داخلية (على سبيل المثال ،كي_أقرب جار) وعندما تحتاج إلى الاحتفاظ بالسمات الرئيسية (على سبيل المثال ، لا يُسمح باستخراج السمات).

السلبية:

- تضيف الخوارزميات الجينية مستوى أعلى من التعقيد إلى التنفيذ الخاص بك وغالبًا لا تستحق ذلك. إذا كان ذلك ممكنًا ، فإن استخدام PCA أو الاستخدام المباشر لخوارزمية مع تحديد الميزات الداخلية يكون أسرع وأسهل.

[1] fitness

طرق التغطية[1]

في طريقة التغطية، يتم اختيار الخصائص من خلال اعتبارها مشكلة بحث يتم فيها تكوين مركبات مختلفة وتقييمها ومقارنتها مع المركبات الأخرى. تقوم هذه الطريقة بتعليم الخوارزمية بشكل متكرر باستخدام مجموعة الميزات الفرعية. استنادًا إلى إخراج النموذج، تتم إضافة الميزات أو طرحها، ومع هذه المجموعة من الميزات، يتم تدريب النموذج مرة أخرى. بعض تقنيات طريقة التغطية هي:

- **الاختيار المقبل:** اختيار العميل المتوقع هو عملية تكرارية تبدأ بمجموعة فارغة من السمات. بعد كل تكرار ، يضيف ميزة ويقيم الأداء لمعرفة ما إذا كان يحسن الأداء. تستمر هذه العملية حتى لا تؤدي إضافة ميزة جديدة إلى تحسين أداء النموذج.
- **الحذف الرجعي:** إزالة الخلفية هي أيضًا طريقة متكررة ، ولكنها عكس الاختيار المقبل. تبدأ هذه التقنية العملية من خلال النظر في جميع الميزات وإزالة الميزة الأقل أهمية. تستمر عملية الحذف هذه حتى لا يؤدي حذف الميزات إلى تحسين أداء النموذج.
- **الاختيار خطوة بخطوة (تدريجي):** الاختيار خطوة بخطوة أو التحديد ثنائي الاتجاه يشبه الاختيار الرائد ، ولكن الاختلاف هو أنه بإضافة ميزة جديدة ، فإنه يفحص أيضًا أهمية ما تمت إضافته مسبقًا الميزات والعثور على أي ميزات محددة مسبقًا غير مهمة.، ما عليك سوى حذف هذه الميزة المعينة عن طريق حذف الخلفية. ومن ثم ، فهو مزيج من الاختيار المقبل والحذف الرجعي.

طريقة التصفية

في طريقة التصفية، يتم تحديد الخصائص بناءً على المعايير الإحصائية. لا تعتمد هذه الطريقة على خوارزمية التعلم وتختار الميزات كخطوة معالجة مسبقة. تقوم طريقة التصفية بتصفية الميزات غير ذات الصلة بالنموذج باستخدام معايير مختلفة من خلال التصنيف. تتمثل ميزة استخدام طرق التصفيةفي أنها تتطلب وقتًا أقل للحساب ولا تؤدي إلى الضبط الزائد للبيانات.

التقليل الخطي: تحليل العنصر الرئيسي (PCA)[2]

يعد تحليل العنصر الرئيسي (PCA) أحد أكثر تقنيات تقليل الأبعاد الخطية شيوعًا والتي تستخدم تحويلًا متعامدًا لتحويل مجموعة من الملاحظات للمتغيرات المرتبطة المحتملة إلى مجموعة من قيم المتغيرات الخطية غير المرتبطة تسمى المكونات الأساسية. قبل تقديم التفاصيل، دعنا نفكرفي السؤال التالي:

---

[1] Wrapper Methods

[2] Principal Component Analysis

للحصول على أمثلةفي مساحة ميزة متعامدة[1]، كيف يمكننا استخدام المستويات الفائقة لعرض العينات؟

حدسيًا، إذا كان مثل هذا المستوى الفائق موجودًا، فمن المحتمل أن يكون له الخصائص التالية:

الحد الأدنى من خطأ إعادة البناء: يجب أن تكون العينات على مسافة قريبة من هذه المستوى الفائق.

التباين الأقصى: يجب أن يكون اسقاط[2] (التأثيرات) للعينات على المستوى الفائق متباعداً.

يجد PCA تمثيلًا للأبعاد للبيانات مع الحفاظ على التغييرات (أي المعلومات المميزة) قدر الإمكان. يقوم PCA بذلك عن طريق معالجة الارتباط بين الميزات. إذا كان الارتباط بين مجموعة فرعية من الميزات كبيرًا جداً، يحاول PCA الجمع بين الخصائص شديدة الارتباط وعرض هذه البيانات مع عدد أقل من الميزات الخطية غير المرتبطة. تستمر الخوارزميةفي تنفيذ تقليل الارتباط هذا، وتجد اتجاهات التباين الأقصىفي بيانات الأبعاد الأصلية، وتعرضهافي مساحة أبعاد أصغر. تُعرف هذه المكونات المشتقة حديثًا بالمكونات الأساسية. يتم تعريف هذا التحويل بحيث يكون للمكون الرئيسي الأول أكبر تباين ممكن (أي أنه يحتوي على أكبر قدر ممكن من التباينفي البيانات) ولكل مكون لاحق بدوره أكبر تباين ممكن. باستخدام هذه المكونات، من الممكن إعادة بناء الميزات الرئيسية (ليس بالكامل). تحاول خوارزمية PCA بنشاط تقليل خطأ إعادة الإعمار أثناء البحث عن المكونات المثلى.

فيما يلي نظرة عامة على كيفية إجراء تحليل المكون الرئيسي في مجموعة بيانات.

الخطوة 1. مجموعة البيانات

لنفترض أن لدينا مجموعة بيانات بها سمات أو متغيرات n يُشار إليها بـ$X_1, X_2, \dots, X_n$. إذا كانت هناك عينات N ، فإن قيم i للخاصية X_i تساوي $X_{i1}, \dots, X_{iN}$ (كما في الجدول أدناه).

الميزات	العينة 1	العينة 2	⋯	العينة N
X_1	X_{11}	X_{12}	⋯	X_{1N}
X_2	X_{21}	X_{22}	⋯	X_{2N}
⋮				
X_i	X_{i1}	X_{i2}	⋯	X_{iN}
⋮				
X_n	X_{n1}	X_{n2}	⋯	X_{nN}

الخطوة 2. حساب متوسط المتغيرات

نحسب متوسط$\bar{X}_i$ للمتغير X_i:

[1] orthogonal feature space

[2] projections

$$\bar{X}_i = \frac{1}{N}(x_{i1} + x_{i2} + \cdots + x_{iN})$$

الخطوة 3. حساب مصفوفة التغاير

ضع في اعتبارك المتغيرات X_i و X_j. يتم تعريف التغاير المرتب للزوج $(X_i\ ,\ X_j)$ على النحو التالي:

$$Cov(X_j\ ,X_i)\ = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ik} - \bar{X}_i)(x_{jk} - \bar{X}_j)$$

نحسب المصفوفة $S_{n\times n}$ والتي تسمى مصفوفة التغاير:

$$S = \begin{bmatrix} Cov(X_1\ ,X_1) & Cov(X_1\ ,X_2) & \cdots & Cov(X_1\ ,X_n) \\ Cov(X_2\ ,X_1) & Cov(X_2\ ,X_2) & \cdots & Cov(X_2\ ,X_n) \\ \vdots & & & \\ Cov(X_n\ ,X_1) & Cov(X_n\ ,X_n) & \cdots & Cov(X_n\ ,X_n) \end{bmatrix}$$

الخطوة 4. حساب القيم الذاتية والمتجهات الذاتية لمصفوفة التغاير

افترض أن S مصفوفة تغاير و كذلك نفس المصفوفة I التي لها أبعاد مشابهة للبعد S..

أ. قم بتكوين المعادلة التالية:

$$\det(S - \lambda I) = 0$$

هذه معادلة كثيرة الحدود من الدرجة n فيλ . n له جذور حقيقية (قد تتكرر بعض الجذور) وهذه الجذور هي قيم خاصة لـ S. نجد الجذور n $\lambda_1, \dots, \lambda_n$ من المعادلة أعلاه.

ب. إذا كانت $\lambda = \lambda'$ قيمة خاصة ، يكون المتجه الخاص للمتجه كما يلي:

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_3 \end{bmatrix}$$

لهذا السبب:

$$(S - \lambda' I)U = 0$$

ثم نجد مجموعة من n المتجهات الذاتية المتعامدة $U_1, \dots, U_n$ ، بحيث يكون U_i متجهًا ذاتيًا يتوافق مع λ_i.

ج. الآن نقوم بتسوية المتجهات الخاصة. بالنظر إلى أي متجه X ، نقوم بتسويته بقسمة X على طوله. طول (أو لينة) المتجه :

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_3 \end{bmatrix}$$

يُعرف كالاتي:

$$\|X\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

فيما يتعلق بأي متجه خاص U ، المتجه الخاص بالتسوية يُحسب:

$$\frac{1}{\|U\|}U$$

ما n بردار ویژه نرمال‌شده $e_1, \dots, e_n$ را توسط

$$e_i = \frac{1}{\|U_i\|}U_i, i = 1,2, \dots, n.$$

الخطوة 5. استخراج مجموعات البيانات الجديدة

رتب القيم الذاتية من اعلى قيمة الى اقل قيمة. اكبر قيمة ذاتية هي المكون الرئيسي الاول.

أ. دع القيم الذاتية تكون بترتيب تنازلي $\lambda_1, \dots, \lambda_n$ ودع المتجهات الذاتية للوحدة المقابلة تكون $e_1, \dots, e_n$.

ب. اختر عددًا صحيحًا p بحيث يكون $1 \le p \le n$.

ج. حدد المتجهات الذاتية المقابلة لقيم $\lambda_1, \dots, \lambda_p$، . وشكل المصفوفة التالية $p \times n$

$$F = \begin{bmatrix} e_1^T \\ e_2^T \\ \vdots \\ e_p^T \end{bmatrix}$$

د. شكل مصفوفة $n \times N$:

$$X = \begin{bmatrix} X_{11} - \bar{X}_1 & X_{12} - \bar{X}_1 & \cdots & X_{1N} - \bar{X}_1 \\ X_{21} - \bar{X}_2 & X_{22} - \bar{X}_2 & \cdots & X_{2N} - \bar{X}_1 \\ \vdots & & & \\ X_{31} - \bar{X}_3 & X_{n2} - \bar{X}_n & \cdots & X_{nN} - \bar{X}_n \end{bmatrix}$$

ه. ثم نحسب المصفوفة التالية:

$$X_{\text{جديد}} = FX$$

لاحظ أن هذه مصفوفة $p \times N$ تعطينا مجموعة بيانات من عينات N مع الخاصية p.

الخطوة 6. مجموعه بيانات جديدة

تعد مصفوفة $X_{\text{جديد}}$ مجموعة بيانات جديدة. يمثل كل صف من هذه المصفوفة قيم الخاصية.

الخطوة 7. النتيجة

بهذه الطريقة ، يساعدنا تحليل المكونات الرئيسية على تقليل حجم مجموعة البيانات. **لاحظ أنه لا يمكن استعادة مجموعة البيانات الأصلية ذات البعد n من مجموعة البيانات الجديدة.**

نوضح فكرة تحليل المكون الرئيسي بمثال. في هذا المثال ، يتم تقديم كافة تفاصيل العمليات الحسابية. هذا لإعطاء القارئ فكرة عن مدى تعقيد العمليات الحسابية وأيضًا لمساعدة القارئ على القيام بحسابات يدوية دون اللجوء إلى حزم البرامج.

مثال. وفقًا للبيانات الواردة في الجدول أدناه ، استخدم PCA لتقليل البعد من 2 إلى 1.

الميزات	العينة 1	العينة 2	العينة 3	العينة 4
X_1	4	8	13	7
X_2	11	4	5	14

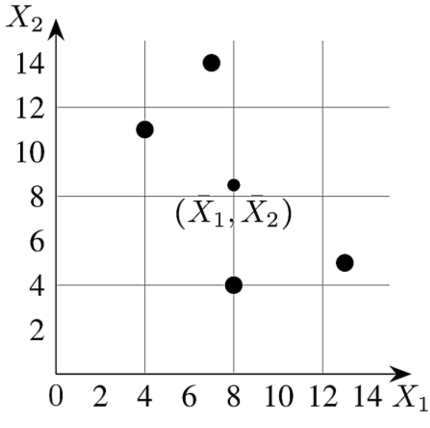
1. مخطط التشتت (التبعثر) للبيانات

لدينا:

$$\bar{X}_1 = \frac{1}{4}(4+8+13+7) = 8$$

$$\bar{X}_1 = \frac{1}{4}(11+4+5+14) = 8.5$$

يوضح الشكل التالي الرسم البياني النقطي للبيانات جنبًا إلى جنب مع النقطة $\bar{X}_1, \bar{X}_2$:



شكل 8ـ12. پراکندگی دادهها

2. حساب مصفوفة التغاير

يتم حساب التغايرات على النحو التالي:

$$Cov(X_1, X_1) = \frac{1}{N-1}\sum_{k=1}^{N}(x_{1k} - \bar{X}_1)^2$$

$$= \frac{1}{3}((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2)$$

$$= 14$$

$$Cov(X_1, X_2) = \frac{1}{N-1}\sum_{k=1}^{N}(x_{1k} - \bar{X}_1)(x_{2k} - \bar{X}_2)$$

$$= \frac{1}{3}((4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5))$$

$$= -11$$

$$Cov(X_2, X_1) = -11$$

$$Cov(X_2, X_2) = \frac{1}{N-1}\sum_{k=1}^{N}(x_{2k} - \bar{X}_2)^2$$

$$= \frac{1}{3}((11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2)$$

$$= 23$$

وفقًا لذلك ، فإن مصفوفة التغاير تساوي:

$$S = \begin{bmatrix} Cov(X_1, X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Cov(X_2, X_2) \end{bmatrix}$$

$$= \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

3. قيم مصفوفة التغاير الذاتية

المعادلة المميزة لمصفوفة التغاير تساوي:

$$0 = \det(S - \lambda I)$$

$$= \begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix}$$

$$= (14-\lambda)(23-\lambda) - (-11)\times(-11)$$

$$= \lambda^2 - 37\lambda + 201$$

لدينا حل المعادلة:

$$\lambda = \frac{1}{2}(37 \pm \sqrt{565})$$

$$= 30.3849, 6.6151$$

$$= \lambda_1, \lambda_2$$

4. حساب المتجهات الذاتية

للعثور على المكون الرئيسي الأول ، نحتاج فقط إلى حساب المتجه الذاتي المقابل لأكبر قيمة ذاتية. في المثال الحالي ، أكبر قيمة ذاتية هي λ_1. لذلك نحسب المتجه الخاص لـ λ_1.

المتجه الخاص لـ$\lambda = \lambda_1$ هو المتجه $U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ الذي يحقق المعادلة التالية:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = (S - \lambda_1 I)X$$

$$= \begin{bmatrix} 14-\lambda_1 & -11 \\ -11 & 23-\lambda_1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$= \begin{bmatrix} (14-\lambda_1)u_1 - 11u_2 \\ -11u_1 + (23-\lambda_1)u_2 \end{bmatrix}$$

وهو ما يعادل المعادلتين التاليتين:

$$(14-\lambda_1)u_1 - 11u_2 = 0$$
$$-11u_1 + (23-\lambda_1)u_2 = 0$$

باستخدام نظرية نظام المعادلة الخطية نجد أن هذه المعادلات ليست مستقلة والإجابات بها:

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda_1} = t$$

يتم تقديمها ، بمعنى أن

$$u_1 = 11t, u_2 = (14-\lambda_1)t$$

حيث t هو أي رقم حقيقي. بأخذ $t = 1$، لدينا المتجه الذاتي المقابل لـ λ_1

$$U_1 = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix}$$

لايجاد وحدة المتجه الذاتي، نحسب طول X_1:

$$\|U_1\| = \sqrt{11^2 + (14-\lambda_1)^2}$$
$$= \sqrt{11^2 + (14-30.3849)^2}$$
$$= 19.7348$$

إذن ، المتجه الذاتي لـ λ_1 يساوي :

$$e_1 = \begin{bmatrix} 11/\|U_1\| \\ 14-\lambda_1/\|U_1\| \end{bmatrix}$$
$$= \begin{bmatrix} 11/19.7348 \\ 14-30.3849/19.7348 \end{bmatrix}$$
$$= \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

من خلال إجراء حسابات مماثلة ، يمكن الحصول على المتجه الذاتي e_2 المقابل للقيمة الذاتية $\lambda = \lambda_1$:

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}.$$

5. حساب المكون الرئيسي الأول

افترض $\begin{bmatrix} X_{1k} \\ X_{2k} \end{bmatrix}$ k_ عينة موجود في جدول البيانات. المكون الرئيسي الأول لهذه العينة هو:

$$e_1^T \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix} = [0.5574 \quad -0.8303] \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix}$$
$$= 0.5574(X_{1k} - \bar{X}_1) - 0.8303(X_{2k} - \bar{X}_2).$$

على سبيل المثال ، يتم حساب المكون الرئيسي الأول للعينة الأولى $\begin{bmatrix} X_{11} \\ X_{21} \end{bmatrix} = \begin{bmatrix} 4 \\ 11 \end{bmatrix}$ على النحو التالي:

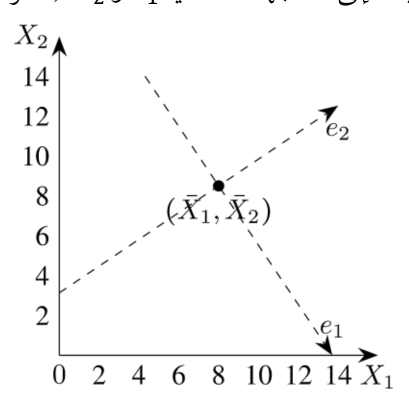
$$[0.5574 \quad -0.8303] \begin{bmatrix} X_{11} - \bar{X}_1 \\ X_{21} - \bar{X}_2 \end{bmatrix}$$
$$= 0.5574(X_{11} - \bar{X}_1) - 0.8303(X_{21} - \bar{X}_2)$$
$$= 0.5574(4 - 8) - 0.8303(11 - 8.5)$$
$$= -4.30535$$

تم تلخيص نتائج الحسابات في الجدول أدناه:

X_1	4	8	13	7
X_2	11	4	5	14
المكون الرئيسي الأول	−4.3052	3.7361	5.6928	**−5.1238**

6. المعنى الهندسي للمكون الرئيسي الأول

بالنسبة للشكل 8ـ12 ، نقدم محاور إحداثيات جديدة. أولاً نغير الأصل إلى "المركز" $\bar{X}_1, \bar{X}_2$ ثم نغير اتجاه محاور الإحداثيات إلى المتجهات الذاتية e_1 و e_2 (انظر الشكل 8ـ13).

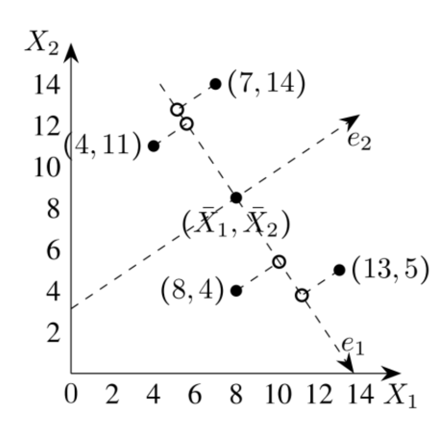


الشكل 8ـ13. جهاز الإحداثيات للمكونات الاصلية

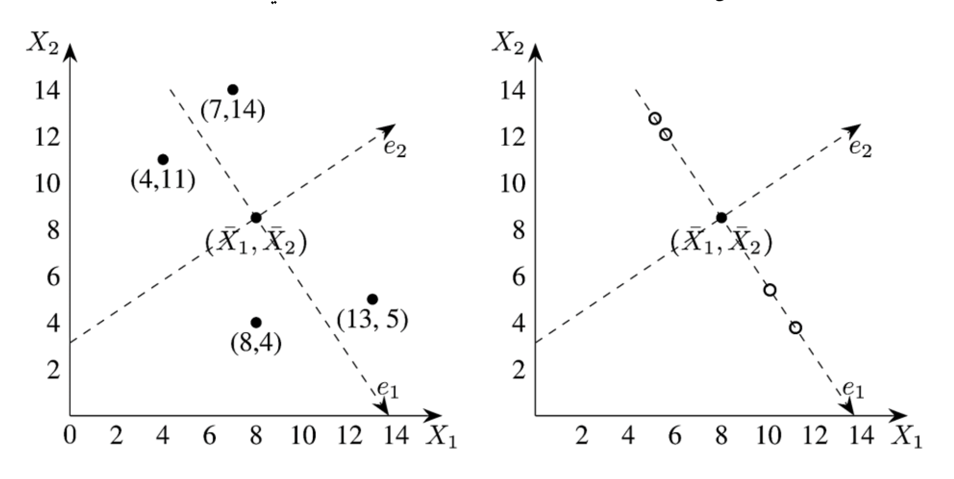
ثم نرسم الخطوط العمودية من النقاط المعطاة إلى المحور e_1 (انظر الشكل 8ـ14). المكون الرئيسي الأول لإحداثيات e_1 هو الأرجل الرأسية. يمكن اعتبار تعيين نقاط البيانات في المحور e_1 بمثابة تقريب لنقاط البيانات المقدمة ، حتى نتمكن من استبدال مجموعة البيانات المقدمة بهذه النقاط.

الآن ، يمكن الإشارة بوضوح إلى كل من هذه التقريبات برقم ، والإحداثيات التقريبية لـ e_1. لذلك ، يمكن تمثيل مجموعة البيانات ثنائية الأبعاد المقدمة تقريبًا بواسطة مجموعة البيانات أحادية البعد التالية (انظر الشكل 8ـ15).:

المكون الرئيسي الاول	−4.3052	3.7361	5.6928	−5.1238



الشكل 8ـ14. تعيين البيانات على محور المكون الرئيسي الأول



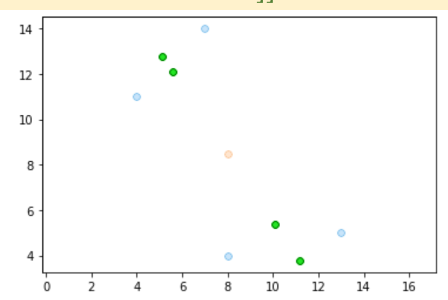
الشكل 8ـ15. التمثيل الهندسي التقريبي أحادي البعد للبيانات

يمكنك أيضًا تشغيل الكود التالي في بايثون:

```
In [1]: #data
        import numpy as np
        import matplotlib.pyplot as plt
        from sklearn.decomposition import PCA
        X = np.array([[4,11],[8,4],[13,5],[7,14]])
```

```
X1=np.mean(X[:, 0])
X2=np.mean(X[:, 1])
pca = PCA(n_components=1)
pca.fit(X)
X_pca = pca.transform(X)
X_new = pca.inverse_transform(X_pca)
plt.scatter(X[:, 0], X[:, 1], alpha=0.2)
plt.scatter(X1, X2, alpha=0.2)
plt.scatter(X_new[:, 0], X_new[:, 1], alpha=0.8)
plt.axis('equal');
print(pca.components_)

Out [1]:    [[ 0.55738997 -0.83025082]]
```



مزايا و عيوب PCA

المزايا:

- **سهل الحساب:** يعتمد PCA على الجبر الخطي ، والذي يتم حله بسهولة حسابيًا عن طريق الكمبيوتر.
- **يزيد من سرعة خوارزميات التعلم الآلي الأخرى:** تتقارب خوارزميات التعلم الآلي بشكل أسرع عند تدريبها على المكون الرئيسي بدلاً من مجموعة البيانات الرئيسية.

العيوب:

- **توحيد البيانات:** يجب توحيد البيانات قبل تنفيذ PCA ، وإلا يصبح من الصعب تحديد المكونات الأساسية المثلى.
- **تفترض PCA علاقة خطية بين الخصائص:** الخوارزمية غير مناسبة للعلاقات غير الخطية.

التعلم المتشعب (التقليل غير الخطى)

في القسم السابق ، رأينا كيفية استخدام تحليل المكون الرئيسي في عمل تقليل الأبعاد. في حين أن PCA مرن وسريع ، إلا أنه لا يعمل بشكل جيد عندما تكون هناك علاقات غير خطية في

البيانات. للتغلب على هذا القيد ، يمكننا الرجوع إلى مجموعة من الأساليب تسمى التعلم المتشعب؛ مجموعة من المقدرين غير الخاضعين للإشراف الذين يسعون إلى وصف مجموعات البيانات على أنها مشعبات منخفضة الأبعاد مدمجة في مساحات عالية الأبعاد.

عندما تفكر في مشعب ، أقترح أن تتخيل ورقة. من الممكن استئجار معدات للعمل على تأجير معدات إلكترونية على الإنترنت ثم ثنيها أو تركيبها. في مصطلحات التعلم المتشعب ، يمكن اعتبار هذه الصفيحة الصخرية متعددة الأبعاد ثنائية الأبعاد مضمنة في الفضاء ثلاثي الأبعاد. لا يؤدي تدوير أو إعادة لف أو شد قطعة من الورق في المساحة التالية إلى تغيير الشكل الهندسي الناعم للورق. إذا قمت بثني الورق أو لفها أو تجعيدها ، فسيظل متشعبًا ثنائي الأبعاد ، لكن التضمين في مساحة ثلاثية الأبعاد لم يعد خطيًا. تسعى خوارزميات التعلم المتشعبة إلى تعلم الطبيعة ثنائية الأبعاد لهذه الورقة ، أو بعبارة أخرى ، لتحديد هيكل هذا المتشعب.

بشكل أساسي ، تنص الفرضية المتشعبة على أن البيانات عالية الأبعاد في العالم الحقيقي يتم وضعها على مشعبات منخفضة الأبعاد مدمجة في الفضاء عالي الأبعاد. بعبارات أبسط ، هذا يعني أن البيانات ذات الأبعاد الأعلى غالبًا ما تكون أقرب بكثير إلى مشعب ذي أبعاد أقل. تسمى عملية النمذجة المتشعبة التي تستند إليها أمثلة التدريب التعلم المتشعب.

تُظهر نظرية الهندسة التفاضلية أنه يمكن اعتبار المشعب مساحة رياضية مجردة يمكن تحديد هندستها الجوهرية بالكامل من خلال المقاييس المحلية ومعلومات الجوار الصغيرة جدًا التي تشبه الفضاء الإقليدي. لذلك قد نعتقد أنه يمكن استخدام بنية الجوار المحلية للمشعب لتضمين نقاط البيانات في مساحة عالية الأبعاد في فضاء خصائص الأبعاد. وبالتالي ، إذا تم تضمين مشعب الأبعاد في مساحة عالية الأبعاد ، فإن العينات التي تبدو معقدة للغاية في الفضاء عالي الأبعاد لها نفس الخصائص المحلية مثل العينات الموجودة في الفضاء الإقليدي. لذلك ، يمكننا إنشاء رسم الخرائط محليًا ثم توسيعه ليشمل المساحة بأكملها. عندما يتم تقليل الأبعاد إلى اثنين أو ثلاثة ، يمكننا بشكل طبيعي تصور البيانات ، لذا فإن تعلم المشعب مفيد لأغراض التوضيح أيضًا. في استمرار لهذا القسم ، سنراجع بعض خوارزميات التعلم المتشعب الشائعة.

ايزوماب[1]

ايزوماب هي خوارزمية تعلم تشعبية تتعلم النموذج الداخلي للبيانات. يربط أقرب الجيران ويشكل رسمًا بيانيًا. ثم يقوم بحساب أقصر مسار بين جميع عقد الرسم البياني. يقدّر المسافة الجيوديسية (مسافة نقطتين من السطح) للنقاط. أخيرًا ، يطبق مقياسًا متعدد الأبعاد على الرسم البياني لمصفوفة تباعد الرسم البياني ، مما يسمح بتضمين البيانات الأصلية بأبعاد منخفضة.

هناك طريقتان عامتان لبناء مخطط الجوار. النهج الأول هو تحديد عدد الجيران. على سبيل المثال ، استخدام كي_أقرب جار مع معيار المسافة الإقليدية. طريقة أخرى هي تعيين عتبة

[1] Isomap

المسافة للنظر في جميع النقاط التي تقل مسافة عن ϵ كجيران. ومع ذلك ، فإن كلا النهجين لهما نفس القيود: إذا كان نطاق الجوار المحدد k أو كبيرًا ، فقد تحدث " دائرة مقصورة" ، حيث تُعتبر بعض النقاط البعيدة عن طريق الخطأ قريبة من بعضها البعض. من ناحية أخرى ، إذا كان نطاق الجوار المحدد صغيرًا جداً ، فقد تحدث "دائرة مفتوحة" يتم فيها فصل بعض المناطق عن بعضها البعض.

تعد خوارزمية فلويد[1] خوارزمية فعالة لحساب أقصر مسار بين كل زوج من الرؤوس في الرسم البياني ، والتي تعمل بشكل أفضل مع الرسوم البيانية الكثيفة (الرسوم البيانية ذات الحواف المتعددة). ومع ذلك ، تُفضل خوارزمية ديكسترا[2] عندما يكون الرسم البياني منفردًا. تحتوي خوارزمية فلويد على تعقيد $O(n^3)$ في أسوأ الأحوال ، بينما تحتوي خوارزمية ديكسترا مع هرم فيبوناتشي على تعقيد $O(Kn^2 logn)$ حيث K هو حجم الجوار.

يوفر ايزوماب فقط إحداثيات منخفضة الأبعاد لعينات التدريب ، ولكن كيف يمكننا إحضار العينات الجديدة إلى الفضاء السفلي؟ تتمثل الطريقة العامة في تعليم نموذج الانحدار باستخدام إحداثيات الأبعاد العليا لعينات التدريب كمدخلات والإحداثيات السفلية المقابلة كإخراج. ثم استخدم نموذج الانحدار المدرب للتنبؤ بالإحداثيات السفلية للعينات الجديدة. مثل هذه الطريقة لا يبدو أنها شائعة ولكن في الوقت الحالي لا يبدو أن هناك حل أفضل !!

التضمين الخطى المحلى [٣] (LLE)

على عكس ايزوماب، التي تحافظ على المسافة بين العينات ، فإن التضمين الخطي المحلي (LLE) يهدف إلى الحفاظ على العلاقات الخطية بين العينات المجاورة. الاقتباس الشائع حول خوارزمية LLE هو "فكر عالميًا ، مناسبًا محليًا"[4]: بعبارة أخرى ، تنظر الخوارزمية إلى الأجزاء الصغيرة المحلية حول كل عينة وتستخدم هذه الأجزاء لعمل متشعب أوسع. كما هو موضح في الشكل 8ـ16 ، افترض أنه يمكن إعادة بناء إحداثيات نقطة عينة x_i من خلال مجموعة خطية من إحداثيات العينات المجاورة x_k, x_j و x_l، أي:

$$x_i = w_{ij}x_j + w_{ik}x_k + w_{il}x_l$$

الغرض من LLE هو الحفاظ على علاقة عالية في الفضاء البعدي.

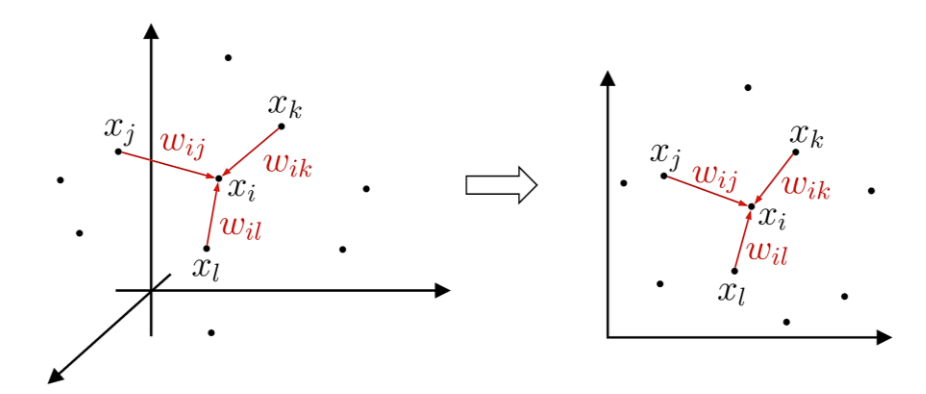
---

[1] Floyd’s algorithm

[2] Dijkstra's algorithm

[3] Locally Linear Embedding

[4] *think globally, fit locally*



الشكل 8ـ16. يتم الحفاظ على العلاقة بين إعادة بناء العينات في الفضاء عالي الأبعاد.

يبدأ LLE بتحديد مؤشرات الجوار Q_i لعينة x_i ثم يجد أوزان إعادة البناء (تجديد) الخطية لعينات w_i في Q_i:

$$min_{w_1,\dots,w_m} \sum_{i=1}^{m} \left\| x_i - \sum_{j\epsilon Q_i} w_{ij} x_j \right\|_2^2$$

$$s.t \sum_{j\epsilon Q_i} w_{ij} x_j = 1$$

حيث x_i و x_j معروفان. اذا كان $C_{jk} = (x_i - x_j)^T (x_i - x_k)$ ، إذن w_{ij} لديه حل مغلق[1] الشكل:

$$w_{ij} = \frac{\sum_{j\epsilon Q_i} C_{jk}^{-1}}{\sum_{l,s\epsilon Q_i} C_{ls}^{-1}}$$

نظرًا لأن LLE، تخزن w_i في مساحة تقليل الأبعاد ، يمكن الحصول على إحداثيات الابعاد z_i من x_i على النحو التالي:

$$min_{z_1,\dots,z_m} \sum_{i=1}^{m} \left\| z_i - \sum_{j\epsilon Q_i} w_{ij} z_j \right\|^2$$

z_i يحسن إحداثيات البعد السفلي x_i .

اذا كان $\boldsymbol{Z} = (z_1, \dots, z_m) \epsilon \mathbb{R}^{\acute{d} \times m}$, $(\boldsymbol{W})_{ij} = w_{ij}$ و $\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{W})^T (\boldsymbol{I} - \boldsymbol{W})$، اذن يمكن صياغة المعادلة السابقة

$$min_Z \;\; tr(\boldsymbol{ZMZ^T})$$
$$s.t\, \boldsymbol{ZZ^T} = 1$$

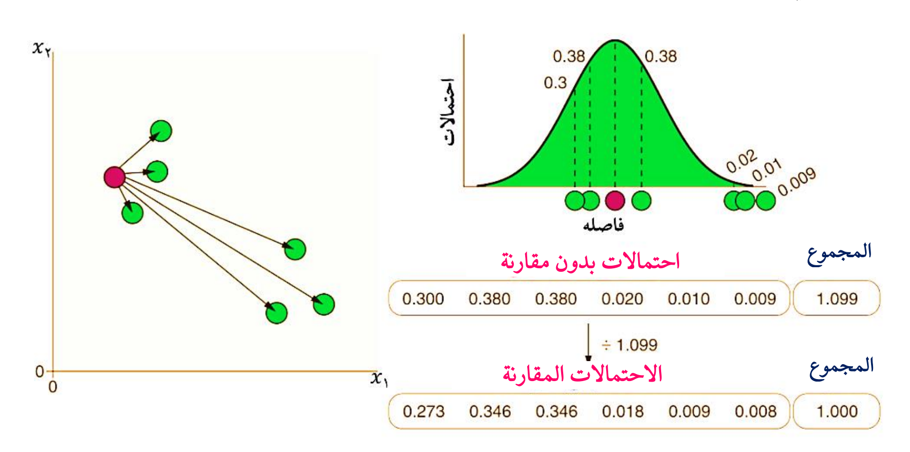
[1] closed form solution

يمكننا حل المعادلة أعلاه عن طريق تحليل القيم الذاتية : **ZT** عبارة عن مصفوفة تتكون من المتجهات الذاتية $\acute{d}$ مع أصغر القيم الذاتية **M**.

تی- تضمين الجوار الموزع عشوائيًا (t-SNE) [1]

يعد تضمين الجوار العشوائي الموزع (t-SNE) أحد أكثر تقنيات تقليل الأبعاد غير الخطية شيوعًا لتصور البيانات عالية الأبعاد. يقوم t-SNE بذلك عن طريق نمذجة أي نقطة عالية الأبعاد في فضاء ثنائي أو ثلاثي الأبعاد ، حيث تكون النقاط المتشابهة قريبة من بعضها البعض والنقاط غير المتشابهة بعيدة. للقيام بذلك ، تقوم t-SNE بعمل توزيعين احتماليين ، أحدهما على أزواج من النقاط في الفضاء عالي الأبعاد والآخر على أزواج من النقاط في مساحة منخفضة الأبعاد ، بحيث تكون النقاط المتشابهة أكثر احتمالية وتقل احتمالية النقاط المتباينة. بتعبير أدق ، يقلل t-SNE من تباعد كولباك ــ ليبلير[2] بين توزيعي الاحتمال.

تتمثل الخطوة الأولى في خوارزمية t-SNE في حساب المسافة بين كل عينة وعينة أخرى في مجموعة البيانات. بشكل افتراضي ، يتم استخدام المسافة الإقليدية لقياس المسافة ، وهي مسافة الخط المستقيم بين نقطتين في مساحة الميزة. ثم يتم تحويل هذه المسافات إلى احتمالات. يمكنك رؤيته في الشكل 8ــ17.



الشكل 8ــ17. حساب المسافات وتحويلها إلى احتمالات.

بالنسبة لعينة معينة في مجموعة البيانات ، يتم قياس المسافة بين هذا العينة والعينات الأخرى. ثم يركز التوزيع الطبيعي على هذه الحالة ، ويتم تحويل المسافات إلى احتمالات عن طريق تعيينها لكثافة احتمالية التوزيع الطبيعي. الانحراف المعياري لهذا التوزيع الطبيعي يرتبط عكسيًا بكثافة العينات حول العينة. بمعنى آخر ، إذا كان هناك العديد من العينات القريبة (أكثر كثافة) ، فإن

---

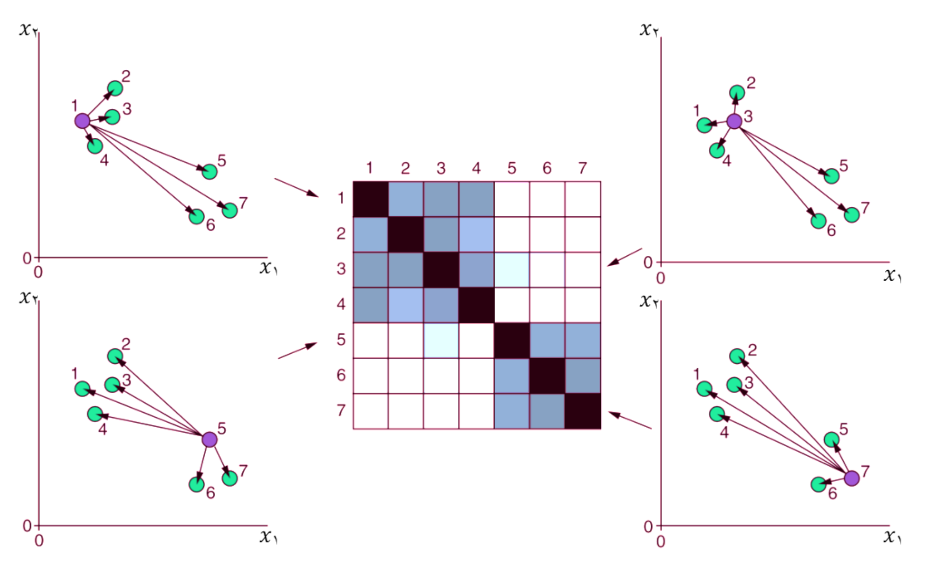
[1] t-distributed stochastic neighbor embedding

[2] Kullback-Leibler divergence

الانحراف المعياري للتوزيع الطبيعي يكون أصغر. ولكن إذا كانت هناك حالات قليلة في المنطقة المجاورة (كثافة أقل) ، فإن الانحراف المعياري يكون أكبر.

بعد تحويل المسافات إلى احتمالات ، يتم قياس الاحتمالات لكل حالة بقسمة مجموعها. يؤدي هذا إلى وصول مجموع الاحتمالات لكل عينة في مجموعة البيانات إلى 1. إن استخدام انحرافات معيارية مختلفة لكثافات مختلفة ثم تسوية الاحتمالات إلى 1 لكل عينة يعني أنه في حالة وجود مجموعات كثيفة ومجموعات فردية من العينات في مجموعة البيانات ، فإن t-SNE يوسع المجموعات الكثيفة ويشتت المجموعات. يمكن توضيحها معًا بسهولة أكبر.

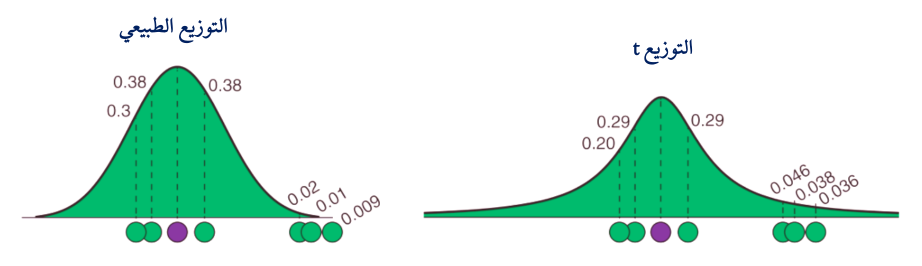
عندما يتم حساب الاحتمالات المقاسة لكل عينة في مجموعة البيانات ، لدينا مصفوفة احتمالية تصف مدى تشابه كل حالة مع بعضها البعض. يظهر هذا الرسم التوضيحي في الشكل 8ـ18 كخريطة حرارية.



الشكل 8ـ18. يتم تخزين الاحتمالات المقاسة لكل عينة كمصفوفة من القيم. هذه موضحة هنا كخريطة حرارية: كلما اقتربت العينتان، كلما كان الصندوق أكثر قتامة، مما يشير إلى بعدهما عن الخريطة الحرارية.

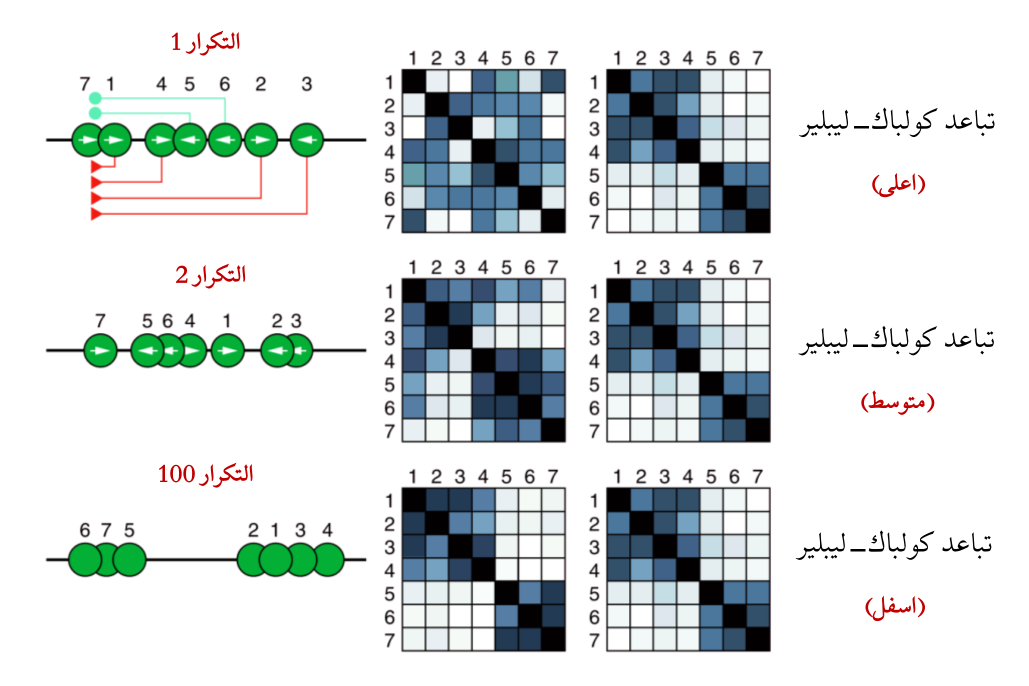
مصفوفة الاحتمالات الخاصة بنا هي الآن مرجعنا أو نمطنا لكيفية ارتباط قيم البيانات ببعضها البعض في الفضاء الرئيسي عالي الأبعاد. تتمثل الخطوة التالية في خوارزمية t-SNE في ترتيب العناصر بشكل عشوائي على محورين (ليس بالضرورة محورين، ولكن هذا هو الحال عادةً) (هذا هو المكان الذي يحصل فيه t-SNE على اسمه العشوائي).

يحسب t-SNE المسافات بين العينات في هذا الفضاء العشوائي والبعد الأدنى الجديد ، ويحولها إلى احتمالات كما كان من قبل. الاختلاف الوحيد هو أنه بدلاً من استخدام التوزيع الطبيعي ، فإنه يستخدم الآن توزيع T-Student. يشبه توزيع T إلى حد ما التوزيع الطبيعي ، باستثناء أن الوسط ليس طويلًا جدًا وأن الجانبين أكثر سلاسة وأكثر سحبًا للخارج (الشكل 8ـ 19).



الشكل 8ـ19. التوزيع الطبيعي وتوزيع t

تتمثل مهمة t-SNE الآن في "تشابك" نقاط البيانات حول هذه المحاور الجديدة خطوة بخطوة لجعل مصفوفة الاحتمالية في مساحة البعد الأدنى قريبة قدر الإمكان من مصفوفة الاحتمال الرئيسية في مساحة البعد الأعلى. الحدس هنا هو أنه إذا كانت المصفوفات متشابهة قدر الإمكان، فإن البيانات في كلا الفراغين تكون قريبة من بعضها البعض. من أجل أن تكون مصفوفة الاحتمالية في الفضاء منخفض الأبعاد مشابهة لمصفوفة الفضاء عالية الأبعاد ، يجب أن تكون كل عينة أقرب إلى العينات التي كانت قريبة منها في البيانات الأصلية وبعيدًا عن العينات التي كانت بعيدة. نتيجة لذلك ، فإن العينات التي يجب أن تكون قريبة تجذب جارها إليها ، ولكن العينات التي يجب أن تكون بعيدة تكون مشتتة. يؤدي توازن هذه القوى الجاذبة والمنافرة إلى تحرك كل عنصر في مجموعة البيانات في اتجاه يجعل مصفوفتي الاحتمال أكثر تشابهًا قليلاً. الآن ، في هذا الموقف الجديد ، تتم إعادة حساب مصفوفة احتمالية البعد السفلي وتتحرك العينات مرة أخرى ، مما يجعل مصفوفات الأبعاد الدنيا والعليا تبدو أكثر تشابهًا قليلاً مرة أخرى. تستمر هذه العملية حتى نصل إلى عدد محدد مسبقًا من التكرارات ، أو حتى يتحسن التباعد (الاختلاف) بين المصفوفات. بمجرد أن تتقارب هذه العملية التكرارية مع اختلاف تباعد كولباك ـ ليبلير المنخفض ، يجب أن يكون لدينا تمثيلات منخفضة الأبعاد لبياناتنا الأصلية التي تحتفظ بأوجه التشابه بين العينات القريبة. العملية برمتها موضحة في الشكل 8ـ20.



الشكل 8ـ20. عملية خوارزمية t-SNE. يتم تهيئة العينات بشكل عشوائي على محاور جديدة (يظهر المحورفي هذا المثال). يتم حساب مصفوفة الاحتمالات لهذا المحور ويتم خلط العناصر لجعلها تبدو مثل المصفوفة الأصلية عالية الأبعاد عن طريق تقليل تباعد كولباك ـ ليبلير. أثناء الإزاحة، تنجذب العينات إلى العينات التي تشبهها (خطوط دائرية) وتبتعد عن الكائنات غير المتشابهة (خطوط مثلثة).

قد تتساءل عن سبب استخدامنا للتوزيع t لتحويل المسافات إلى احتمالات في الفضاء ذي الأبعاد المنخفضة؟ الرجوع إلى الشكل 8ـ19 مرة أخرى لفهم أفضل. كلا جانبي توزيع t أوسع من التوزيع الطبيعي. هذا يعني أنه من أجل الحصول على نفس الاحتمال للتوزيع الطبيعي ، يجب أن تبتعد الحالات غير المتشابهة عن الحالة التي يتركز فيها توزيع t. يساعد هذا في تفريق مجموعات البيانات التي قد تكون موجودة في البيانات ويساعدنا في التعرف عليها بسهولة أكبر. ومع ذلك ، فإن الاستنتاج المهم هو أن t-SNE غالبًا ما يُقال إنه يحافظ على البنية المحلية في تمثيل منخفض الأبعاد ، ولكنه عادة لا يحافظ على الهيكل العالمي. في الممارسة العملية ، هذا يعني أنه يمكننا تفسير العينات القريبة من بعضها البعض في التمثيل النهائي على أنها متشابهة ، لكن لا يمكننا بسهولة تحديد مجموعات العينات المشابهة لمجموعات العينات الأخرى في البيانات الأصلية.

يتم عرض خطوات خوارزمية t-SNE في أدناه:

الخطوة 1. حساب المسافة بين أزواج من النقاط.

أولاً ، يقارن t-SNE أزواج من نقاط البيانات في الفضاء عالي الأبعاد. يشير الاحتمال الشرطي لكل زوج إلى مدى قرب النقاط. يمكن أن يكون مقياس التشابه المستخدم لبناء توزيع الاحتمالات هو المسافة الإقليدية. وفقًا لذلك ، تكون نقاط البيانات القريبة محتملة للغاية ، بينما تكون نقاط البيانات المنفصلة بشكل كبير صغيرة جدًا. رياضياً ، يكون الاحتمال الشرطي بين نقطتي البيانات x_i و x_j كما يلي:

$$p_{j|i} = \frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k\neq i} \exp\left(\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}\right)}$$

حيث σ_i^2 هو تباين التوزيع الغاوسي. تجد الخوارزمية قيمة σ_i بسبب **ارباك**[1] هذا التوزيع. يقيس **الارباك** إمكانية التنبؤ ببعض التوزيعات الاحتمالية وهو عامل مهم في t-SNE. يمكن للمستخدم تحديد هذا المعامل، وتتراوح القيم المفيدة من 5 إلى 50.

كثافة الاحتمال لزوج من النقاط تتناسب مع تشابهها. بالنسبة لنقاط البيانات القريبة ، ستكون $p_{j|i}$ مرتفعة نسبيًا ، وبالنسبة للنقاط المنفصلة بشكل كبير ، ستكون $p_{j|i}$ صغيرة. تناظر الاحتمالات الشرطية في مساحة البعد العلوي للحصول على أوجه التشابه النهائية في مساحة البعد الأعلى. لقياس التشابه الزوجي بين نقطتي بيانات ، تكون الاحتمالات الشرطية متماثلة مع احتمالات p_{ij} الشائعة عن طريق حساب متوسط نقطتين ، مع الأخذ في الاعتبار إجمالي نقاط N:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{N}$$

الخطوة 2. بناء فضاء البعد السفلي.

بعد قياس احتمالات أزواج النقاط لمساحة البيانات الرئيسية ، فإن الخطوة التالية هي إنشاء مساحة ذات أبعاد منخفضة. يتم حساب أوجه التشابه بين q_{ij} بين نقطتي البيانات y_i و y_j في مساحة منخفضة الابعاد على النحو التالي:

$$p_{j|i} = \frac{(1 + \left\|y_i - y_j\right\|^2)^{-1}}{\sum_{k\neq l}(1 + \left\|y_i - y_j\right\|^2)^{-1}}$$

---

[1] perplexity

y_i و y_j هي نظائر منخفضة الأبعاد لنقاط البيانات عالية الأبعاد x_i و x_j.

الخطوة 3. تقليل الفرق بين توزيعات احتمالية الفضاء عالية الأبعاد والفضاء الجديد.

تتمثل الخطوة التالية في تقليل الاختلاف بين التوزيع الاحتمالي للفضاء الأصلي عالي الأبعاد والمساحة الجديدة التي تم إنشاؤها عن طريق تقليل عدد الأبعاد. يتم استخدام دالة الخطأ لتقليل هذا الاختلاف مع الانحدار الاشتقاقي. في حالة t-SNE ، هذه الدالة هي دالة خطأ تباعد كولباك–ليبلير (KL):

$$D_{KL}(p_{ij}||p_{ji}) = \sum_{i \neq j} p_{ij} log \frac{p_{ij}}{q_{ij}}$$

تعمل t-SNE على تحسين النقاط في مساحة البعد السفلي باستخدام الانحدار الاشتقاقي ودالة فقدان تباعد KL. لماذا نستخدم KL؟ عندما نقوم بتقليل تباعد KL ، يصبح q_{ij} عمليًا هو نفسه p_{ij} ، وبالتالي فإن بنية البيانات في مساحة الأبعاد العليا ستكون مشابهة لهيكل البيانات في مساحة الأبعاد السفلية. بناءً على معادلة الاختلاف KL:

- إذا كانت قيمة p_{ij} كبيرة ، فنحن بحاجة إلى قيمة كبيرة لـ q_{ij} لإظهار النقاط المحلية ذات التشابه العالي.
- إذا كانت قيمة p_{ij} صغيرة ، فنحن بحاجة إلى قيمة صغيرة لـ q_{ij} لإظهار النقاط المحلية المتباعدة.

الخطوة 4. استخدم توزيع t لحساب التشابه بين نقطتين في مساحة البعد الأدنى.

يستخدم t-SNE توزيع t-student لحساب التشابه بين نقطتين في مساحة البعد السفلي بدلاً من التوزيع الغوسي. ينشئ توزيع نقطة الإنطلاق التوزيع الاحتمالي للنقاط في مساحة البعد الأدنى ويساعد على تقليل مشكلة الازدحام.

هذه الطريقة لها أيضًا عيوبها. التطبيق المباشر على مجموعات البيانات متعددة الأبعاد غير عملي ، لأن المسافات متشابهة جداً في الأبعاد الكبيرة. لذلك ، قبل استخدام t-SNE ، يمكن تطبيق البيانات متعددة الأبعاد على طرق تقليل الأبعاد الأخرى أولاً. بعد ذلك ، يمكن تطبيق t-SNE إذا لزم الأمر. يعد التسوية خطوة مهمة قبل إدخال البيانات ، حيث أن مقياس المسافة المستخدمة هو المسافة الإقليدية. نظرًا لأن الخوارزمية عشوائية ، يمكن تشغيلها عدة مرات للعثور على الحل الأقل ضررًا بسبب اختلاف KL.

UMAP

في السنوات الأخيرة ، تم اقتراح طريقة جديدة لتقليل الأبعاد تسمى UMAP. هذه التقنية غير خطية وتتبع فكرة t-SNE لإنشاء مساحة بُعد أقل للبيانات. ومع ذلك ، هناك العديد من

الاختلافات الرئيسية. يعتمد نموذجه الرياضي على الهندسة الريمانية والطوبولوجيا. لا يستخدم UMAP التسوية في التوزيعات الاحتمالية. في المقابل ، يتم استخدام نسخة متجانسة من كي– أقرب جار. يتمتع UMAP بالعديد من المزايا على t-SNE من حيث الأداء. يكون أسرع عند تضمينه في فضاء بعد جديد أكبر من ثلاثة أبعاد. يحافظ UMAP بشكل أفضل على الهيكل العام أثناء تخزين المعلومات المحلية. وفقًا للخوارزمية ، ترتبط كل نقطة في العرض عالية الأبعاد بجارتها. بينما يحتوي t-SNE على معامل مهم (ارتباك) ، فإن UMAP يحتوي على الكثير. أهم المعاملات هي عدد الجيران ، والحد الأدنى للمسافة ، وعدد المكونات والقياس. ومع ذلك، فإن الميزة هي أن هذه المعاملات مفهومة بشكل مباشر وسنشرحها بمزيد من التفصيل.

يعد معامل عدد أقرب الجيران أحد أهم المعاملات. يحدد هذا المعامل حجم الجيران المحليين المقصودين ، مما يؤثر على بناء الرسم البياني الأولي ذي الأبعاد العالية. يتم الاحتفاظ بالتفاصيل المحلية عند تحديد قيم صغيرة لهذه المعامل. في المقابل ، مع القيم العالية ، تولي الخوارزمية الانتباه إلى الهيكل العام في البيانات الأولية. يمكن اعتبار هذا المعامل بمثابة تغيير من معامل الارتباك في t-SNE.

المعامل الثاني المهم هو الحد الأدنى للمسافة التي تتحكم في مدى احتواء النقاط معًا في عرض منخفض الأبعاد. تؤدي القيم المنخفضة إلى تمثيلات أكثر كثافة ، بينما تؤدي القيم العالية إلى تخصيص أقل كثافة لنقاط البيانات. يحدد المعامل رقم المكون أبعاد مساحة البعد المصغر التي يتم عرض البيانات عليها. يتحكم المعامل المتري في كيفية حساب المسافة في المساحة المحيطية لبيانات الإدخال.

دالة الخسارة التي يستخدمها UMAP هي إنتروبيا التفاعل بين التمثيلات الطوبولوجية للأبعاد العالية والمنخفضة للمسافات. يعد هذا أيضًا فرقًا مهمًا مقارنةً بـ t-SNE ، والذي يستخدم اختلاف KL كدالة خسارة. ومع ذلك ، مثل العديد من تقنيات الأبعاد ، يستخدم UMAP الانحدار الاشتقاقي لتقليل دالة الخسارة هذه.

عيب تقنية تقليل الأبعاد هذه أنه ليس من المهم اختيار مجموعة جيدة من المعاملات. قد تكون المسافات بين المجموعات بلا معنى ، حيث يتم أخذ المسافات المحلية في الاعتبار عند إنشاء الرسم البياني. تعد خوارزمية UMAP عشوائية وتنتهي في الحد الأدنى المحلي ، لذلك من الأفضل الحصول على نتائج باستخدام نفس المعاملات عدة مرات. يستخدم UMAP العشوائية لتسريع خطوات التقدير وللمساعدة في حل مشكلات التحسين الصعبة. هذا يعني أن تطبيقات UMAP المختلفة يمكن أن تؤدي إلى نتائج مختلفة. UMAP مستقر نسبيًا. لذلك ، يجب أن يكون التباين بين الأداء من الناحية المثالية صغيرًا نسبيًا ، ولكن قد يختلف الأداء. يمكن ضبط وضع الحبوب العشوائي لضمان إمكانية إعادة إنتاج النتائج بدقة.

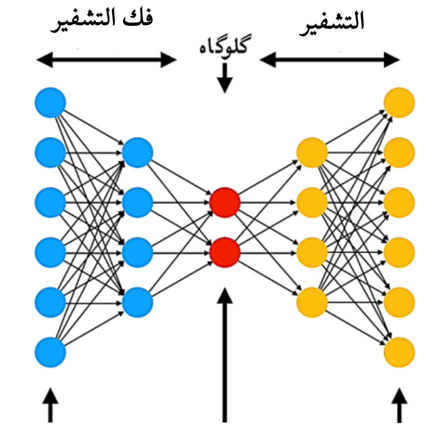
المشفر التلقائى

المشفر التلقائي (Autoencoder) عبارة عن شبكات عصبية تستخدم لتقليل الأبعاد. الهدف من الترميز التلقائي هو تعلم التمثيل المكثف لبيانات الإدخال من خلال تحقيق إعادة البناء عند المخرجات. ببساطة ، المشفرات التلقائية لديها القدرة على اكتشاف التمثيلات عالية الأبعاد لبيانات عالية الأبعاد وقادرة على إعادة إنتاج المدخلات من المخرجات التي تحاول نقل المدخلات x إلى نفسها. قد يبدو تعلم النسخ من المدخلات إلى المخرجات تافهاً ، ولكن بفرض بعض القيود على بنية الشبكة ، يضطر المشفر التلقائي إلى تعلم أهم ميزات بيانات التدريب ، حيث لا يمكنه نمذجة كل شيء.

تكمن الفكرة وراء المشفر التلقائي في تلقي بيانات إدخال عالية الأبعاد من خلال عنق زجاجة المعلومات ، وفي هذه العملية ، تحويلها إلى تمثيل منخفض الأبعاد ، ثم إعادة إنتاج بيانات الإدخال من هذا التمثيل الذي تم تعلمه. من خلال تعيين المدخلات في مساحة ذات أبعاد مخفضة ، تكون الشبكة العصبية قادرة على التعلم واستخراج الميزات المختلفة.

الآن ، قد يُطرح السؤال للقارئ حول سبب إزعاجنا لتعلم تمثيل المدخلات الأصلية فقط لإعادة إنتاج المخرجات بأفضل ما نستطيع. الإجابة هي أنه عندما يكون لدينا مدخلات مع العديد من الميزات ، فإن إنشاء تمثيل مضغوط من خلال الطبقات المخفية للشبكة العصبية يمكن أن يساعد في ضغط مدخلات عينة التدريب. لذلك عندما تقوم الشبكة العصبية بمراجعة جميع بيانات التدريب وضبط وزن جميع عقد الطبقة المخفية ، فإن ما يحدث هو أن الأوزان تمثل حقًا نوع الإدخال الذي نراه عادةً. نتيجة لذلك ، إذا أردنا إدخال نوع آخر من البيانات ، مثل الحصول على بيانات مع بعض الضوضاء ، فستكون شبكة المشفر التلقائي قادرة على اكتشاف الضوضاء عند الإدخال وإزالة جزء على الأقل من الضوضاء عند توليد الإخراج..

يتكون المشفر التلقائي من جزأين من الشبكة العصبية ، **المشفر وجهاز فك التشفير**. يعمل المشفر على تقليل حجم مجموعة البيانات الكبيرة إلى مجموعة صغيرة ، بينما يقوم جهاز فك التشفير بشكل أساسي بتوسيع البيانات منخفضة البيانات إلى البيانات الكبيرة. الغرض من هذه العملية هو محاولة إعادة إنتاج المدخلات الأصلية. إذا كانت الشبكة العصبية جيدة الإنشاء ، فهناك فرصة جيدة لإعادة إنتاج المدخلات الأصلية من البيانات المشفرة. يظهر هيكل وحدة المشفر التلقائي في الشكل 8ـ21.



الشكل 8ـ21. هيكل المشفر التلقائي

في الممارسة العملية ، يمكن استخدام المشفر التلقائي لتقليل حجم الميزة ، على غرار الطرق الأخرى ، ولكن أداءها أقوى. لأن نماذج الشبكة العصبية يمكنها استخراج ميزات جديدة بشكل أكثر كفاءة. من الناحية الهيكلية ، تتشابه المشفرات التلقائية إلى حد كبير مع شبكات امامية التغذية ، في أبسط أشكالها ، لديهم طبقة إدخال وطبقة مخفية وطبقة إخراج. يتمثل الاختلاف الهيكلي الرئيسي مع شبكات امامية التغذية في أن المشفرات التلقائية لها نفس عدد العقد في طبقات الإدخال والإخراج الخاصة بها.

ومع ذلك ، فإن الاختلاف الرئيسي بين المشفر التلقائي وشبكة امامية التغذية يكمن في عملية التعلم ؛ تستخدم المشفرات التلقائية البيانات غير المسماة بدلاً من البيانات المصنفة. هناك عدة أنواع من المشفرات التلقائية. سنصف أولاً الإصدار القياسي ثم ننظر إلى الإصدارات الأخرى.

أحد التطبيقات الشائعة للمشفر التلقائى هو تقليل الأبعاد. عادة ما تكون أبعاد التمثيل المضمنة أصغر بكثير من أبعاد الإدخال الأصلية ، وإذا كان المشفر يعمل بشكل جيد ، فإن التمثيل المضمن يعمل كإصدار مضغوط جيد من الإصدار الأصلى بأبعاد إدخال أصغر بكثير.

المشفر التلقائى القياسى

المشفر التلقائي القياسي عبارة عن هيكل يتكون من جزأين. الجزء الأول عبارة عن برنامج تشفير يتلقى بيانات الإدخال ويتعلم ضغطها للحصول على تشفير (يُعرف أيضًا باسم الرمز أو التمثيل الكامن[1]). يمكن القيام بذلك من خلال التعرف على أهم ميزات البيانات. رياضيا ، يمكن تمثيل رمز الطبقة المخفية والمدخلات x على النحو التالي:

[1] latent representation

$$h = \sigma(Wx + b)$$

حيث h هو التمثيل الكامن ، هي دالة تنشيط الشبكة ، b و W هما وزن المصفوفة والتحيز ، على التوالي.

الجزء الثاني هو وحدة فك التشفير التي تستقبل البيانات المشفرة. الغرض من المشفر هو إعادة بناء التشفير بطريقة تجعل التكاثر الناتج أقرب ما يمكن إلى بيانات الإدخال المشفرة. يمكن إعادة إنتاج $\acute{x}$ في هذه الحالة باستخدام

$$\acute{x} = \acute{\sigma}(\acute{W}h + \acute{b})$$

$\acute{\sigma}$، $\acute{W}$ و $\acute{b}$ ي معاملات مختلفة لدالة التنشيط ومصفوفة الوزن والتحيز.

إذا أشرنا إلى المشفر بدالة مثل $h = f(x)$ ووحدة فك التشفير بدالة مثل $r = g(h)$، فيمكن عندئذٍ وصف المشفر بأكمله بدالة $r = g(f(x))$ ، حيث يكون الإخراج r مشابهًا للمدخل الرئيسي x. الغرض من تدريب المشفر التلقائي هو الحصول على h مفيد يمكن أن يمثل مدخلات x بشكل جيد. نموذج المشفر الذاتي المثالي يأخذ في الاعتبار النقاط التالية:

- إنه حساس بدرجة كافية لإدخاله ليكون قادرًا على التكاثر بدقة.
- إنه ليس حساس بما يكفي للإدخال ، بحيث لا يتذكر النموذج ببساطة بيانات التدريب. بعبارة أخرى ، لا يتناسب أكثر من اللازم.

يفرض هذا التوازن على النموذج الاحتفاظ فقط بتغييرات البيانات اللازمة لإعادة إنتاج المدخلات دون الحفاظ على التكرار في المدخلات. في معظم الحالات ، يتضمن ذلك إنشاء دالة خسارة. تُستخدم دالة الخسارة لتقييم الفرق بين القيمة المتوقعة للنموذج والقيمة الفعلية ، وكلما انخفضت الخسارة ، كان أداء النموذج أفضل. يتم تعريف دالة الخسارة للمشفر الذاتي على أنها $L(x, g(f(x)))$ ، حيث L هي دالة الخسارة لحساب الفرق بين x و $g\ (f\ (x))$. الهدف من التدريب على التشفير الذاتي هو تقليل وظيفة الخسارة (فقدان التكاثر). يمكن تدريب الشبكات باستخدام تقنيات الشبكات العميقة القياسية مثل الانتشار الخلفي.

من خلال تدريب المشفر الذاتي الذي كان أداؤه جيدًا في إعادة إنتاج بيانات الإدخال ، نأمل أن يوضح التمثيل العشوائي لـ h بعض الميزات المفيدة في البيانات. لتجنب الحلول غير المهمة وتعلم الميزات المفيدة ، نحتاج إلى إضافة بعض القيود إلى المشفر الذاتي.

يمكن استخدام المشفر الذاتي لاستخراج ميزات مفيدة عن طريق إجبار h على أن يكون أصغر من x. يسمى التشفير الذاتي الذي يكون بعده الكامن أقل من بُعد الإدخال بالتشفير الذاتي غير الكامل. يؤدي تعلم التمثيل غير الكامل إلى إجبار عامل التشفير على تسجيل أبرز سمات بيانات التدريب. بمعنى آخر ، التمثيل الكامن h هو تمثيل موزع يحصل على إحداثيات على طول العوامل الرئيسية للتغيير في البيانات. هذا مشابه للطريقة التي يتم بها عرض خرائط (تأثيرات)

المكونات الرئيسية للتغيير في البيانات. في الواقع ، إذا كانت هناك طبقة مخفية خطية وتم استخدام معيار متوسط الخطأ التربيعي لتدريب الشبكة ، فإن الوحدات المخفية تتعلم تمثيل المدخلات في نطاق المكونات الأساسية الأولى. إذا كانت الطبقة المخفية غير خطية ، فإن المشفر التلقائي يتصرف بشكل مختلف مع PCA ويمكنه تصوير جوانب متعددة الأوجه لتوزيع المدخلات.

خيار آخر هو الحد من h على بُعد أكبر من x. يسمى التشفير الذاتي الذي تكون أبعاده الكامنة أكبر من بُعد الإدخال **بالتشفير الذاتي الكامل**[1]. ومع ذلك ، نظرًا لحجمها الكبير ، تميل أجهزة التشفير وأجهزة فك التشفير إلى الضبط الزائد. ومن ثم ، في مثل هذه الحالات ، يمكن حتى لجهاز التشفير وفك التشفير الخطي تعلم نسخ المدخلات إلى المخرجات دون تعلم أي تمثيلات مفيدة لتوزيع البيانات. لحسن الحظ ، لا يزال بإمكاننا اكتشاف بُنية مثيرة للاهتمام من خلال فرض قيود أخرى على الشبكة. أحد القيود الأكثر استخدامًا هو قيد التشتت في h. المشفر التلقائي الكامل مع قيد التشتت يسمى **المشفر التلقائي الانفرادي**[2].

المشفر التلقائي مع ازالة الضوضاء

المشفر التلقائي لإزالة الضوضاء يختلف عن المشفر التلقائي القياسي. إشارة الإدخال تالفة جزئيًا في البداية في المشفر التلقائي لإلغاء الضوضاء ثم تغذيتها إلى الشبكة. يتم تدريب الشبكة بطريقة يتم بها استرداد دفق بيانات الإدخال من البيانات السيئة نسبيًا. يسمح هذا للمشفر بفهم البنية الأساسية لإشارات الإدخال من أجل إعادة إنتاج متجه الإدخال الأصلي بشكل مناسب ، لإعادة إنتاج متجه الإدخال الأصلي بشكل كافٍ. كما ذكرنا سابقًا ، عادةً ما تقلل المشفرات التلقائية من دالة الخسارة L. يقلل المشفر التلقائي لازالة الضوضاء من دالة الخسارة التالية:

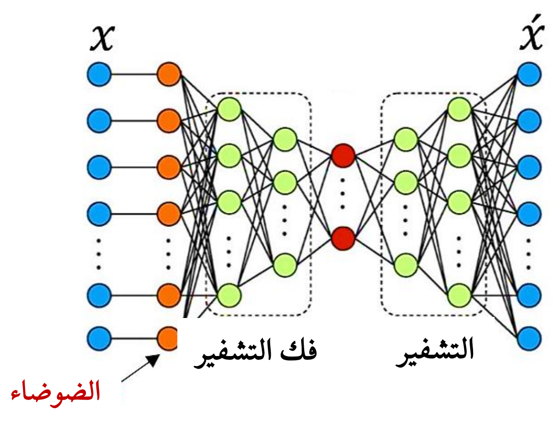
$$L(x, g(f(\acute{x})))$$

حيث $\acute{x}$ نسخة من x تالفة بالضوضاء. تظهر آلية المشفر التلقائي لإزالة الضوضاء في الشكل 8ـ 22.

---

[1] overcomplete autoencoder

[2] sparse autoencoder



الشكل 8ـ22. المشفر التلقائي لإزالة الضوضاء

تتم إضافة الضوضاء إلى الإدخال x أثناء التدريب للتأكد من أن المشفر التلقائي يتعلم ميزات توزيع البيانات المفيدة..

المشفر التلقائي الانفرادي

يمكن للمشر التلقائي الانفرادي تعلم الهياكل القيمة في بيانات الإدخال عن طريق فرض التشتت على الوحدات المخفية أثناء التدريب. يمكن تحقيق التشتت بشكل عام عن طريق إضافة تعبيرات إضافية (Ω) إلى دالة الخسارة أثناء التدريب أو عن طريق إعادة ضبط جميع الوحدات المخفية يدويًا باستثناء بعض الأساسيات.:

$$L\left(x, g\left(f(x)\right)\right) + \Omega(h)$$

من خلال معاقبة دالة الخسارة في الطبقات المخفية ، يؤدي ذلك إلى تنشيط عدد قليل فقط عند إدخال عينة جديدة. الحدس الكامن وراء هذا النهج هو ، على سبيل المثال ، إذا ادعى المرء التخصص في الرياضيات وعلوم الكمبيوتر وعلم النفس والفيزياء والكيمياء ، فقد يتعلم المرء معرفة سطحية جداً بهذه الموضوعات. ومع ذلك ، إذا ادعى أنه متخصص فقط في علوم الكمبيوتر، فنحن نرغب في الحصول على رؤى مفيدة منه ، وينطبق الشيء نفسه على المشفر التلقائي الذي نقوم بتدريبه ؛ يضمن عدد أقل من العقد التي لا تزال تحافظ على دوالها عند تنشيطها أن المشفر التلقائي تتعلم بالفعل التمثيل المخفي بدلاً من المعلومات الإضافية في بيانات الإدخال.

أحد التطورات في مجال المشفر التلقائي الانفرادي هو المشفر التلقائي k انفرادي. في هذا المشفر التلقائي ، نختار k الخلايا العصبية ذات دوال التنشيط الأعلى ونتجاهل دوال التنشيط الأخرى باستخدام دوال تنشيط ReLU وتعيين العتبة للعثور على أكبر الخلايا العصبية. يضبط هذه القيمة k للحصول على أفضل مستوى مبعثر لمجموعة البيانات.

المشفر التلقائي المكدسي[1]

بعض مجموعات البيانات لها علاقة معقدة بالخصائص. لذلك ، لا يكفي استخدام مشفر تلقائي واحد فقط. وذلك لأن المشفر التلقائي الفردي قد لا يكون قادرًا على تقليل حجم خصائص الإدخال. لذلك في مثل هذه الحالات ، نستخدم المشفر التلقائي للمكدس. المشفر التلقائي المكدسي ، كما يوحي الاسم ، هي مشفرات تلقائية متعددة مكدسة فوق بعضها البعض.

تقليل الابعاد في بايثون

PCA في بايثون

مجموعة البيانات

في حين أن معظم مجموعات البيانات الواقعية مثل الصور والبيانات النصية كبيرة جدًا ، فإننا نستخدم مجموعة البيانات الرقمية المكتوبة بخط اليد MNIST للتبسيط. مجموعة بيانات MNIST عبارة عن مجموعة من الصور الرمادية لأرقام مكتوبة بخط اليد بين 0 و 9 تحتوي على 60.000 صورة بحجم 28 × 28 بكسل. لذلك ، تحتوي مجموعة البيانات هذه على 60000 عينة بيانات بأبعاد 784. لتقليل الأبعاد في مجموعة البيانات هذه، نستخدم PCA لتقليل أبعاد البيانات وعرض البيانات في مساحة الميزات منخفضة الأبعاد. يرسم هذا المثال البيانات مع 784 ميزة في مساحة الميزة ثنائية الأبعاد ويوضح النتائج.

استيراد المكتبات

```
In [1]: from keras.datasets import mnist, fashion_mnist
        import time
        import numpy as np
        import pandas as pd
        from sklearn.decomposition import PCA
        import matplotlib.pyplot as plt
        from matplotlib import colors as mcolors
        import seaborn as sns
```

استيراد مجموعة البيانات

```
In [2]: (X_train, y_train) , (X_test, y_test) = mnist.load_data()
```

[1] Stacked Autoencoder

تحضير البيانات

قم بإنشاء مصفوفة بعدد الصور وعدد البكسل في الصورة وانسخ بيانات X_train إلى X.

```
In [4]: X = np.zeros((X_train.shape[0], 784))
        for i in range(X_train.shape[0]):
            X[i] = X_train[i].flatten()
```

نقوم بتقسيم مجموعة البيانات وجمع 25٪ من بيانات تدريب MNIST وتخزينها في إطار بيانات.

```
In [4]: X = pd.DataFrame(X)
        Y = pd.DataFrame(y_train)
        X = X.sample(frac=0.25,
        random_state=1400).reset_index(drop=True)
        Y = Y.sample(frac=0.25,
        random_state=1400).reset_index(drop=True)
        df = X
```

استخدام PCA في مجموعة البيانات MNIST

```
In [5]: pca = PCA(n_components=2)

        pca_results = pca.fit_transform(df.values)
```

ينشئ PCA بُعدين ، المكون الرئيسي 1 والمكون الرئيسي 2. نضيف مكونين من PCA جنبًا إلى جنب مع تسمياتهم في إطار البيانات. العلامات مطلوبة للتمثيل المرئي فقط.

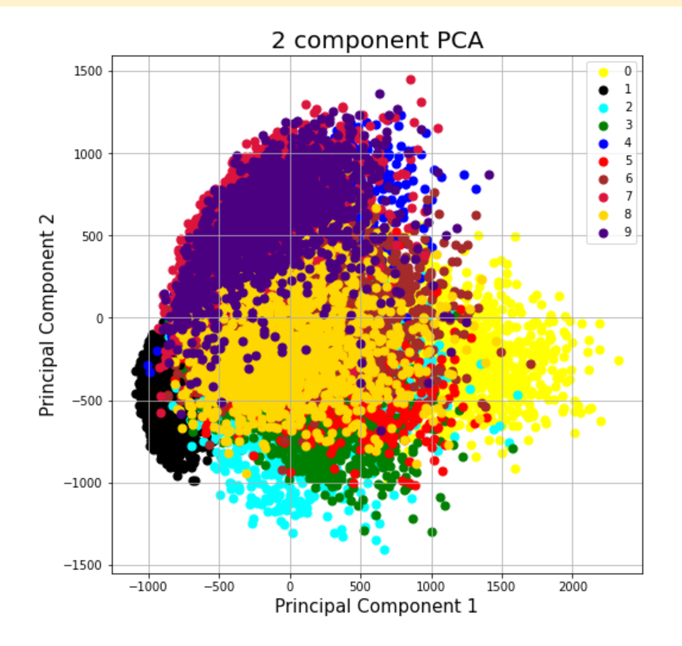
```
In [1]: pca_df = pd.DataFrame(data = pca_results
                    , columns = ['pca_1', 'pca_2'])
        pca_df['label'] = Y
```

التمثيل المرئي

في هذه الخطوة ، قمنا بتدريب الخوارزمية الخاصة بنا وقمنا ببعض التنبؤات. الآن نريد أن نرى مدى دقة خوارزمية لدينا.

```
In [3]: fig = plt.figure(figsize = (8,8))
        ax = fig.add_subplot(1,1,1)
        ax.set_xlabel('Principal Component 1', fontsize = 15)
        ax.set_ylabel('Principal Component 2', fontsize = 15)
        ax.set_title('2 component PCA', fontsize = 20)
        targets = [0,1,2,3,4,5,6,7,8,9]
        colors = ['r', 'g', 'b']
        colors = dict(mcolors.BASE_COLORS,
        **mcolors.CSS4_COLORS)
```

```
colors=['yellow', 'black', 'cyan', 'green', 'blue', 'red',
'brown','crimson', 'gold', 'indigo']
for target, color in zip(targets,colors):
    indicesToKeep = pca_df['label'] == target
    ax.scatter(pca_df.loc[indicesToKeep, 'pca_1']
               , pca_df.loc[indicesToKeep, 'pca_2']
               , c = color
               , s = 50)
ax.legend(targets)
ax.grid()
```



t-SNE في بايثون

استيراد المكتبات

```
In [1]: from sklearn.manifold import TSNE
        from keras.datasets import mnist
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D
        import seaborn as sns
        import matplotlib.patheffects as PathEffects
        import numpy as np
        import pandas as pd
```

استيراد مجموعة البيانات

In [2]:
```
(X_train, y_train) , (X_test, y_test) = mnist.load_data()
```

تحضير البيانات

In [4]:
```
X = np.zeros((X_train.shape[0], 784))
for i in range(X_train.shape[0]):
    X[i] = X_train[i].flatten()
```

نقوم بتقسيم مجموعة البيانات وجمع 80٪ من بيانات تدريب MNIST وتخزينها في إطار بيانات.

In [4]:
```
X = pd.DataFrame(X)
Y = pd.DataFrame(y_train)
X = X.sample(frac=0.80,
random_state=10).reset_index(drop=True)
Y = Y.sample(frac=0.80,
random_state=10).reset_index(drop=True)
df = X
df['label'] = Y
```

استخدام t-SNE في مجموعة البيانات MNIST

In [5]:
```
tsne = TSNE(n_components=2, verbose=1,
perplexity=40, n_iter=300)
tsne_results = tsne.fit_transform(df)
```

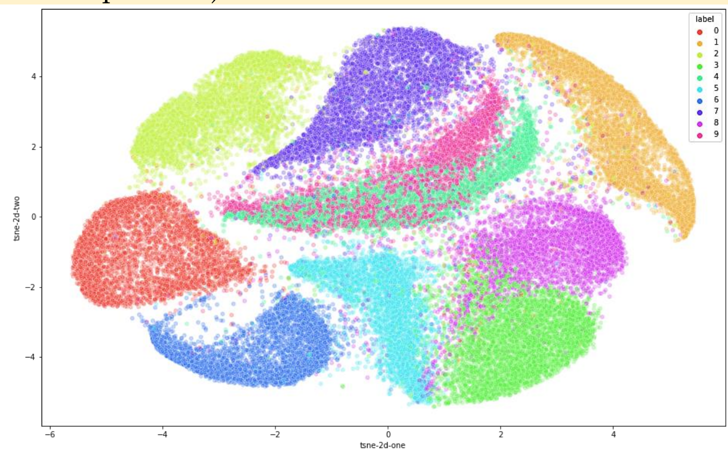
out [5]:
```
[t-SNE] Computing 121 nearest neighbors...
[t-SNE] Indexed 48000 samples in 0.026s...
[t-SNE] Computed neighbors for 48000 samples in 139.468s...
[t-SNE] Computed conditional probabilities for sample 1000 / 48000
[t-SNE] Computed conditional probabilities for sample 2000 / 48000
[t-SNE] Computed conditional probabilities for sample 3000 / 48000
[t-SNE] Computed conditional probabilities for sample 4000 / 48000
[t-SNE] Computed conditional probabilities for sample 5000 / 48000
[t-SNE] Computed conditional probabilities for sample 6000 / 48000
[t-SNE] Computed conditional probabilities for sample 7000 / 48000
[t-SNE] Computed conditional probabilities for sample 8000 / 48000
[t-SNE] Computed conditional probabilities for sample 9000 / 48000
[t-SNE] Computed conditional probabilities for sample 10000 / 48000
[t-SNE] Computed conditional probabilities for sample 11000 / 48000
[t-SNE] Computed conditional probabilities for sample 12000 / 48000
[t-SNE] Computed conditional probabilities for sample 13000 / 48000
[t-SNE] Computed conditional probabilities for sample 14000 / 48000
[t-SNE] Computed conditional probabilities for sample 15000 / 48000
[t-SNE] Computed conditional probabilities for sample 16000 / 48000
[t-SNE] Computed conditional probabilities for sample 17000 / 48000
[t-SNE] Computed conditional probabilities for sample 18000 / 48000
[t-SNE] Computed conditional probabilities for sample 19000 / 48000
[t-SNE] Computed conditional probabilities for sample 20000 / 48000
[t-SNE] Computed conditional probabilities for sample 21000 / 48000
[t-SNE] Computed conditional probabilities for sample 22000 / 48000
[t-SNE] Computed conditional probabilities for sample 23000 / 48000
[t-SNE] Computed conditional probabilities for sample 24000 / 48000
[t-SNE] Computed conditional probabilities for sample 25000 / 48000
[t-SNE] Computed conditional probabilities for sample 26000 / 48000
```

```
[t-SNE] Computed conditional probabilities for sample 27000 / 48000
[t-SNE] Computed conditional probabilities for sample 28000 / 48000
[t-SNE] Computed conditional probabilities for sample 29000 / 48000
[t-SNE] Computed conditional probabilities for sample 30000 / 48000
[t-SNE] Computed conditional probabilities for sample 31000 / 48000
[t-SNE] Computed conditional probabilities for sample 32000 / 48000
[t-SNE] Computed conditional probabilities for sample 33000 / 48000
[t-SNE] Computed conditional probabilities for sample 34000 / 48000
[t-SNE] Computed conditional probabilities for sample 35000 / 48000
[t-SNE] Computed conditional probabilities for sample 36000 / 48000
[t-SNE] Computed conditional probabilities for sample 37000 / 48000
[t-SNE] Computed conditional probabilities for sample 38000 / 48000
[t-SNE] Computed conditional probabilities for sample 39000 / 48000
[t-SNE] Computed conditional probabilities for sample 40000 / 48000
[t-SNE] Computed conditional probabilities for sample 41000 / 48000
[t-SNE] Computed conditional probabilities for sample 42000 / 48000
[t-SNE] Computed conditional probabilities for sample 43000 / 48000
[t-SNE] Computed conditional probabilities for sample 44000 / 48000
[t-SNE] Computed conditional probabilities for sample 45000 / 48000
[t-SNE] Computed conditional probabilities for sample 46000 / 48000
[t-SNE] Computed conditional probabilities for sample 47000 / 48000
[t-SNE] Computed conditional probabilities for sample 48000 / 48000
[t-SNE] Mean sigma: 451.266121
[t-SNE] KL divergence after 250 iterations with early exaggeration: 95.580673
[t-SNE] KL divergence after 300 iterations: 4.694413
```

التمثيل المرئي

الآن بعد أن أصبح لدينا البعدين الناتج ، يمكننا تصورهما عن طريق إنشاء مخطط نقطي للبعدين وتلوين كل عينة بالتسمية المقابلة.

```
In [3]: df['tsne-2d-one'] = tsne_results[:,0]
        df['tsne-2d-two'] = tsne_results[:,1]
        plt.figure(figsize=(16,10))
        sns.scatterplot(
            x="tsne-2d-one", y="tsne-2d-two",
            hue=df['label'],
            palette=sns.color_palette("hls", 10),
            data=df,
            legend="full",
            alpha=0.3)
```



النماذج الانتاجية

النماذج المولدة (الانتاجية) هي فئة من نماذج التعلم الآلي المستخدمة لوصف كيفية إنشاء البيانات. لتدريب نموذج إنتاجي ، نقوم أولاً بجمع كميات كبيرة من البيانات في كل منطقة ثم نقوم بتدريب نموذج لإنشاء أو إنتاج بيانات من هذا القبيل. بعبارة أخرى ، تتمتع النماذج الإنتاجية بالقدرة على تعلم إنشاء بيانات مشابهة للبيانات التي نغذيها بها.

يوجد نوعان أساسيان من النماذج في التعلم الآلي: النماذج الإنتاجية[1] والنماذج التمييزية[2]. يتعلم النموذج المميز $P(Y|X)$ ، وهي العلاقة الشرطية بين المتغير الموضوعي Y وميزات X ، وهي طريقة لفرز العلاقة بين المتغيرات. في المقابل ، فإن الهدف من النموذج الإنتاجي هو وصف الاحتمال الكامل لمجموعة البيانات. بمعنى آخر ، إنها طريقة لتعلم أي نوع من توزيع البيانات. يُستخدم النموذج الإنتاجي في التعلم الآلي غير الخاضع للاشراف كوسيلة لوصف الظواهر في البيانات وتمكين أجهزة الكمبيوتر من فهم العالم الحقيقي.

لنفترض أن لدينا مشكلة تعلم خاضعة للإشراف ، حيث x_i هي الخصائص المعطاة لنقاط البيانات و y_i هي العلامات ذات الصلة. إحدى الطرق للتنبؤ بـ y هي تعلم الدالة $f()$ من (x_i, y_i) ؛ بحيث يستقبل x ويعيد y الأكثر احتمالية كناتج. تندرج هذه النماذج في فئة النماذج التمييزية ، لأنك تتعلم كيفية التمييز بين x من الفئات المختلفة. تندرج طرق مثل شبكات متجهات الدعم والشبكات العصبية وأشجار القرار في هذه الفئة. ومع ذلك ، حتى إذا كان بإمكانك تصنيف البيانات بدقة شديدة ، فليس لديك أي فكرة عن كيفية إنشاء البيانات.

الطريقة الثانية هي كيفية إنشاء البيانات وتعلم الدالة $f(x, y)$ التي تشير إلى التكوين المحدد بواسطة x و y معًا. يمكننا بعد ذلك توقع y لـ x الجديدة بإيجاد y الذي يزيد الدرجة $f(x, y)$.

لفهم هذين النموذجين بشكل أفضل ، دعونا نعطي مثالاً. تخيل x كصورة و y كحيوان مثل قطة في الصورة. يخبرنا الاحتمال المكتوب بالصيغة $p(y|x)$ عن مدى اعتقاد النموذج بوجود قطة ، بالنظر إلى صورة الإدخال مقارنة بجميع الاحتمالات التي يعرفها عنها. تسمى الخوارزميات التي تحاول نمذجة تعيين الاحتمالية مباشرة نماذج تمييزية. من ناحية أخرى ، تحاول النماذج الإنتاجية تعلم دالة تسمى معامل الاحتمال $p(y, x)$. يمكننا أن نقرأ كيف يعتقد النموذج أن x صورة وأن هناك قطة y موجودة فيها في نفس الوقت. هذان الاحتمالان مرتبطان ويمكن كتابتهما كـ $p(y, x) = p(x)p(y|x)$. في هذا الصدد ، تشير $p(x)$ إلى مدى احتمالية أن تكون x هي مدخلات الصورة. عادة ما يسمى الاحتمال $p(x)$ دالة الكثافة.

---

[1] Generative models

[2] Discriminative models

يرجع السبب الرئيسي لتسمية هذه النماذج كمولدات إلى حقيقة أن النموذج يمكنه الوصول إلى إمكانية الإدخال والإخراج المتزامن. بناءً على ذلك ، يمكننا إنتاج صور للحيوانات عن طريق أخذ عينات من أنواع مختلفة من الحيوانات y والصور الجديدة x من $p(y,x)$. ومن ثم ، فإن النماذج المولدة لها ميزة مثيرة للاهتمام على النماذج التمييزية ، أي أن لديها القدرة على فهم وشرح الهيكل الأساسي لبيانات الإدخال ؛ حتى في حالة عدم توفر علامات.

بشكل عام ، يعلم النموذج المميز نموذجًا للتوزيع الشرطي مباشرةً ، على سبيل المثال، $p(y,x)$ حيث y هو متغير مستهدف و x عبارة عن مدخلات. في المقابل ، يتعلم النموذج الإنتاجي التوزيع المشترك لـ $p(y,x)$ ويقال إنه قادر على إنشاء حالات توزيع البيانات.

إذا كان النموذج قادرًا حقًا على إنتاج أمثلة جديدة تتبع ظهور كائنات في العالم الحقيقي، فيمكن في الواقع القول إنه تعلم وفهم مفهومًا بدون تدريب. ومن ثم ، تندرج هذه المجموعة من النماذج في فئة النماذج غير الخاضعة للإشراف (يمكن أيضًا تضمين النماذج الإنتاجية في فئة النماذج الخاضعة للإشراف الذاتي).

تعمل هذه النماذج بشكل عام على الشبكات العصبية ويمكنها بشكل طبيعي اكتشاف السمات المميزة للبيانات. تتلقى الشبكات العصبية هذا الفهم الأساسي لبيانات العالم الحقيقي ثم تستخدمه لنمذجة البيانات التي تشبه بيانات العالم الحقيقي.

الغرض الرئيسي من مجموعة متنوعة من النماذج الإنتاجية هو معرفة التوزيع الفعلي لبيانات مجموعة التدريب بحيث يمكن إنشاء نقاط بيانات جديدة مع التعديلات. لكن ليس من الممكن للنموذج أن يتعلم التوزيع الدقيق لبياناتنا ، ولذا فإننا نصمم توزيعًا مشابهًا لتوزيع البيانات الفعلي. للقيام بذلك ، نستخدم معرفة الشبكة العصبية للتعلم الدالي الذي يمكنه تقريب توزيع النموذج إلى التوزيع الفعلي.

انواع النماذج الانتاجية

الهدف من إنشاء النماذج الانتاجية هو معرفة دالة كثافة الاحتمال $p(x)$. تصف كثافة الاحتمالية هذه بشكل فعال سلوك بيانات التدريب لدينا وتمكننا من إنشاء بيانات جديدة عن طريق أخذ عينات من التوزيع. من الناحية المثالية ، نريد أن يتعلم نموذجنا كثافة الاحتمال $p(x)$ التي هي نفسها كثافة البيانات $p_{data}(x)$. هناك عدة استراتيجيات لتحقيق هذا الهدف.

يمكن للمجموعة الأولى من النماذج أن تحسب **صراحة**[1] دالة الكثافة p أو تحاول تقريبها. أي بعد التدريب، يمكننا إدخال نقطة بيانات x في النموذج ، ويعطينا النموذج صلاحية نقطة

[1] explicitly

البيانات، وهي حاصل ضرب $p(x)$. يشار إلى هذه النماذج على أنها نماذج انتاجية صريحة. الفئة الثانية، والمعروفة باسم النماذج الانتاجية الضمنية ، لا تحسب $p(x)$. ومع ذلك ، بعد التدريب على النموذج ، يمكننا أخذ عينة من التوزيع الأساسي.

النماذج المتغيرة الكامنة

تكمن المشكلة الرئيسية في التعلم الآلي في تعلم التوزيع الاحتمالي المعقد $p(x)$ مع مجموعة محدودة من نقاط البيانات ذات الأبعاد x المستمدة من هذا التوزيع. على سبيل المثال ، لمعرفة توزيع الاحتمالات على صور القطط ، نحتاج إلى تحديد التوزيع الذي يمكنه نمذجة الارتباطات المعقدة بين جميع وحدات البكسل التي تشكل كل صورة. النمذجة المباشرة لهذا التوزيع مهمة صعبة أو حتى مستحيلة.

بدلاً من النمذجة المباشرة $p(x)$ ، يمكننا تقديم متغير كامن z وتعريف التوزيع الشرطي $p(x|z)$ للبيانات ، وهو ما يسمى تمثيل. في العلاقات الاحتمالية ، يمكن تفسير z على أنه متغير عشوائي مستمر. على سبيل المثال ، في صور القطط ، يمكن أن تحتوي z على تمثيل كامن لنوع القط أو لونه أو شكله.

وبالتالي ، بوجود z ، يمكننا تقديم توزيع أمامي $p(z)$ على المتغيرات الكامنة لحساب التوزيع المشترك على المتغيرات المرصودة والكامنة:

$$p(x,z) = p(x|z)p(z)$$

يتيح لنا هذا التوزيع المشترك حل توزيع $p(x)$ المعقد بطريقة أبسط.

للحصول على توزيع البيانات $p(x)$ علينا تهميش المتغيرات الكامنة:

$$p(x) = \int p(x,z)dz = \int p(x,z)p(z)dz \qquad (1-8)$$

بالإضافة إلى ذلك ، باستخدام نظرية بايز ، يمكننا حساب التوزيع اللاحق $p(z|x)$ على النحو التالي:

$$p(z|x) = \frac{p(x,z)p(z)}{p(x)} \qquad (2-8)$$

يسمح لنا التوزيع اللاحق باستنتاج المتغيرات الكامنة من المشاهدات.

بالنظر إلى هذه الفكرة ، لدينا الآن المصطلحات الأساسية التالية:

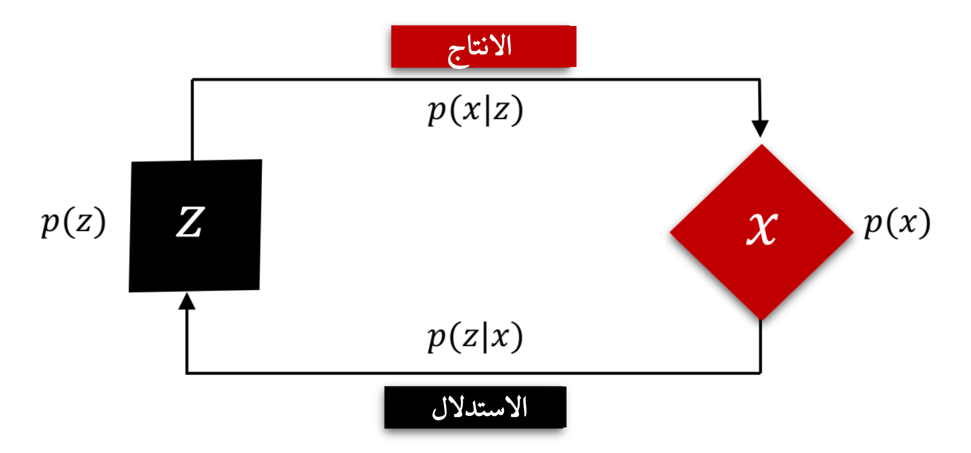
- **التوزيق السابق** $p(z)$: نموذج لسلوك المتغيرات الكامنة.
- **الاحتمالية**$p(x|z)$: يحدد كيفية تعيين المتغيرات الكامنة لنقاط البيانات.
- **التوزيع المشترك** $p(x,z) = p(x|z)p(z)$: الضرب الاحتمالي والسابق ويصف نموذجنا بشكل أساسي.

- **التوزيع الهامشي**$p(x)$: هو توزيع البيانات الأصلية وهو الهدف النهائي للنموذج. يخبرنا التوزيع الهامشي عن مدى إمكانية إنشاء نقطة بيانات.
- **التوزيع اللاحق** $p(z|x)$: يصف المتغيرات المخفية التي يمكن إنشاؤها بواسطة نقطة بيانات محددة.

لاحظ أننا لم نستخدم أي شكل من أشكال التسمية!!

بناءً على ذلك، يمكن تعريف مصطلحين آخرين:

- **الانتاج**: يشير إلى عملية حساب نقطة البيانات x من المتغير الكامن z. في الأساس ، نحن ننتقل من المساحة الكامنة إلى التوزيع الفعلي للبيانات. رياضيا ، يتم الإشارة إلى هذا بواسطة $p(x \mid z)$.
- **الاستدلال**: إن عملية العثور على المتغير المخفي z هي من نقطة البيانات x وتتم صياغتها بواسطة التوزيع اللاحق $p\ (z \mid x)$. من الواضح أن الاستدلال هو معكوس الإنتاج والعكس صحيح. بصريا ، يمكننا النظر في الشكل التالي:



هذا هو المكان الذي يجتمع فيه كل شيء. بافتراض أننا نعرف الاحتمال $p\ (x \mid z)$ ، والسابق $p\ (z \mid x)$ ، والهوامش $p\ (x)$ ، و واللاحق $p\ (z)$ ، يمكننا القيام بما يلي:

- **الانتاج**: لإنشاء نقطة بيانات ، يمكننا أخذ عينة z من $p(z)$ ثم نقطة البيانات $p(x|z)$:

$$z \sim p(z)$$
$$x \sim p(x|z)$$

- **الاستدلال**: من ناحية أخرى ، لاستنتاج متغير كامن ، نقوم بأخذ عينة x من $p(x)$ ثم z من $p\ (z|x)$:

$$x \sim p(x)$$
$$z \sim p(z|x)$$

تتمثل ميزة النماذج ذات المتغيرات الكامنة في قدرتها على التعبير عن عملية الإنتاج التي يتم من خلالها إنشاء البيانات. بشكل عام ، هذا يعني أنه إذا أردنا إنشاء نقطة بيانات جديدة ، يجب علينا أولاً الحصول على عينة $z \sim p(z)$ ثم استخدامها لتجربة ملاحظة جديدة x للتوزيع الشرطي $p\ (x|z)$. أثناء القيام بذلك ، يمكننا أيضًا تقييم ما إذا كان النموذج يوفر تقريبًا جيداً لتوزيع بيانات $p(x)$. يمكنك التفكير في المتغيرات الكامنة باعتبارها عنق الزجاجة التي يجب أن تمر من خلالها جميع المعلومات اللازمة لإنشاء البيانات. نحن نعلم من الفرضية المتشعبة أن البيانات عالية الأبعاد موجودة في المشعبات منخفضة الأبعاد المضمنة في هذا الفضاء عالي الأبعاد. هذه المساحة الكامنة تبرر الأبعاد السفلية. بعبارات أبسط ، المتغيرات الكامنة هي تحويل نقاط البيانات إلى مساحة مستمرة ذات أبعاد أقل. حدسيًا ، تصف المتغيرات الكامنة البيانات بطريقة أبسط.

لاحظ أن التكامل في المعادلة (8ـ1) **ليس له حل تحليلي**[1] لمعظم البيانات التي نتعامل معها ويجب علينا استخدام طريقة للاستدلال لاحقًا في المعادلة (8ـ2).

الاستدلال اللاحق

يعكس التوزيع اللاحق $p(z|x)$ ، وهو مكون رئيسي في الاستدلال الاحتمالية ، يعبر عن معتقداتنا حول المتغيرات الكامنة بعد ملاحظة نقطة بيانات جديدة. ومع ذلك ، غالبًا ما يكون الأخير غير قابل للحل لبيانات العالم الحقيقي ، لأنه لا يوجد حل تحليلي للتكامل في المعادلة (8ـ1) الذي يظهر في مقام المعادلة (8ـ2). هناك طريقتان لتقريب هذا التوزيع. هناك طريقة لأخذ العينات تسمى طريقة سلسلة مونت كارلو ماركوف. ومع ذلك ، فإن هذه الأساليب باهظة الثمن من الناحية الحسابية ولا تتناسب مع مجموعات البيانات واسعة النطاق. الطريقة الثانية هي تقنيات تقريب محددة. إحدى هذه التقنيات هي **الاستدلال القابل للتغيير**[2] ، والذي يستخدم في **المشفر التلقائي القابل للتغيير** ويقوم بذلك عن طريق تحويله إلى مشكلة تحسين. لاحظ ، مع ذلك ، أن عيب هذه الطريقة هو أنها لا يمكن أن تنتج نتائج دقيقة حتى في وقت حسابي غير محدود.

المشفر التلقائي القابل للتغيير

المشفر التلقائي القابل للتغيير هو إطار عام لتعلم النماذج المتغيرة الكامنة مع الاستدلال المتغير. بشكل عام ، يستخدم المشفر التلقائي القابل للتغيير الشبكات العصبية للنموذج الإنتاجي بالإضافة إلى نموذج الاستدلال.

---

[1] no analytical solution

[2] variational inference

لنفترض أنني أعتبر الملاحظات متغيرات عشوائية مستقلة بنفس التوزيع (i.i.d) وافترض أن البيانات يتم إنشاؤها بواسطة بعض المتغيرات العشوائية الكامنة z. من وجهة نظر متغير الاستدلال ، نفترض أن $p(z)$ يمكن ترويضها. نحاول بعد ذلك تعظيم الأدلة لنموذج $p_\theta(x)$ من خلال تقريب التوزيع الكامن المقدّر:

$$\log p_\theta(x) = D_{KL}(q_\varphi(z|x)||p_\theta(z|x)) + \mathcal{L}(\theta,\varphi;x)$$

حيث D_{KL}هو اختلاف Kolbeck-Labler ، و هي المعامل القابل للتغيير، و هو المعامل الإنتاجي ، و L هي الحد الأدنى للمتغير أو الحد الأدنى للملاحظات (ELBO). يمكن إعادة كتابتها وفقًا لنظرية بايز:

$$\mathcal{L}(\theta,\varphi;x) = -D_{KL}(q_\varphi(z|x)||p_\theta(z)) + \mathbb{E}_{q_\varphi(z|x)}[logp_\theta(x|z)]$$

عندما نختار وزن الشبكة العصبية كمعامل ، فهناك علاقة واضحة بين المعادلة أعلاه ومعاملات المشفرات الذاتية القابلة للتغيير. يعمل عنصر اختلاف KL كمنظم على المشفر لتوقع توزيع المتغير الكامن المقدر ، بينما يقلل العنصر الثاني من خطأ الانتاج تحت المتغير الكامن z. لكن النقطة المهمة هي أن المنشور لا يمكن أن يمر عبر الطبقة العشوائية. لأن الاشتقاق غير ممكن بسبب معاملات المتغير φ. بمعنى آخر ، لا يمكن نشر التدرجات للخلف من خلال المتغير الكامن z. تنشأ هذه المشكلة لأن الناشر لا يمكن أن يتدفق عبر العقد العشوائية ويتوقع الناشر قيمًا محددة لتحديد هذه المعاملات. ومن ثم ، يقترح إجراء إعادة تحديد المعايير لحل هذه المشكلة. أي أننا نعيد تحديد معاملات المتغير الكامن z بتحويل مشتق $g_\varphi(\epsilon,x)$ مع متغير ضوضاء إضافي ϵ بالنسبة إلى غاوسي ، يمكن التعبير عن إعادة ضبط المعاملات على النحو التالي:

$$z = \mu + \sigma\epsilon, \quad \epsilon\sim\mathcal{N}(0,I)$$

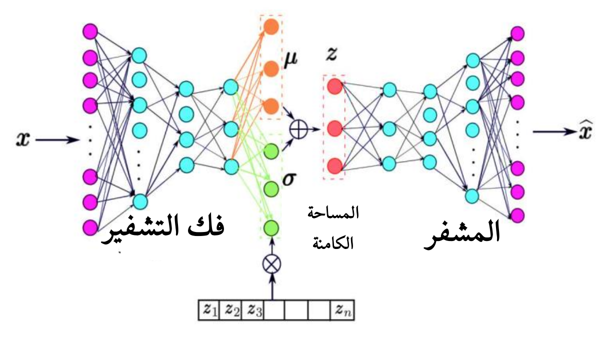
أخيرًا ، يمكن حساب اختلاف KL بشكل تحليلي ، وغالبًا ما يتم استخدام MSE كخسارة إعادة توليد. لذلك ، يتم الحصول على دالة الخسارة للمشفرات التلقائية (VAE) على النحو التالي:

$$\mathcal{L}_{VAE} = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{J}(1+\log(\sigma_{i,j}^2) - \mu_{i,j}^2 - \sigma_{i,j}^2) + \sum_{i=1}^{M}||x_i - \tilde{x}_i||_2^2$$

حيث M هو عدد عينات البيانات ، j هو بُعد البيانات المشفرة ، x_i و $\tilde{x}_i$ هما المعطيات الرئيسية والاستنساخ ، على التوالي. رسم تخطيطي للمشفر التلقائي القابل للتغيير في الشكل 8-22 .[1]

---

[1] لمزيد من المعلومات حول التشفير التلقائي القابل للتغيير، يمكنك الرجوع إلى المرجع التالي:
میلاد وزان، یادگیری عمیق: اصول، مفاهیم و رویکردها، ویرایش نخست، تهران، میعاد اندیشه، ۱۳۹۹



الشكل 8ـ22. المشفر التلقائي القابل للتغيير.

المشفر التلقائى القابل للتغيير فى بايتون

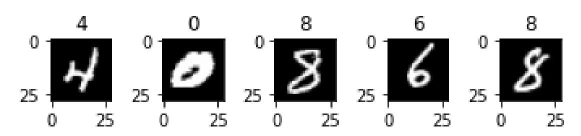
استيراد المكتبات

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
        import seaborn as sns
        import warnings
        import tensorflow
        tensorflow.compat.v1.disable_eager_execution()
```

استيراد مجموعة البيانات

```
In [2]: from tensorflow.keras.datasets import mnist
        (trainX, trainy), (testX, testy) = mnist.load_data()
```

```
In [2]: for j in range(5):
            i = np.random.randint(0, 10000)
            plt.subplot(550 + 1 + j)
            plt.imshow(trainX[i], cmap='gray')
            plt.title(trainy[i])
        plt.show()
```



تحضير البيانات

```
In [4]:    train_data = trainX.astype('float32')/255
           test_data = testX.astype('float32')/255

           train_data = np.reshape(train_data, (60000, 28, 28, 1))
           test_data = np.reshape(test_data, (10000, 28, 28, 1))

           print (train_data.shape, test_data.shape)

out [4]:   (60000, 28, 28, 1) (10000, 28, 28, 1)
```

بناء المشفر

في هذا القسم ، نحدد مشفر نموذج VAE الخاص بنا.

```
In [5]:    import tensorflow

           input_data = tensorflow.keras.layers.Input(shape=(28, 28, 1))

           encoder = tensorflow.keras.layers.Conv2D(64, (5,5),
           activation='relu')(input_data)
           encoder = tensorflow.keras.layers.MaxPooling2D((2,2))(encoder)

           encoder = tensorflow.keras.layers.Conv2D(64, (3,3),
           activation='relu')(encoder)
           encoder = tensorflow.keras.layers.MaxPooling2D((2,2))(encoder)

           encoder = tensorflow.keras.layers.Conv2D(32, (3,3),
           activation='relu')(encoder)
           encoder = tensorflow.keras.layers.MaxPooling2D((2,2))(encoder)

           encoder = tensorflow.keras.layers.Flatten()(encoder)
           encoder = tensorflow.keras.layers.Dense(16)(encoder)
```

التوزيع الكامن وأخذ العينات

هذا القسم مسؤول عن أخذ الميزات الالتفافية من الجزء الأخير وحساب المتوسط والتباين في الميزات الكامنة (كما هو مفترض ، تتبع الميزات الكامنة التوزيع الطبيعي القياسي ويمكن تمثيل التوزيع بالمتوسط والتباين).

```
In [5]:  def sample_latent_features(distribution):
             distribution_mean, distribution_variance = distribution
             batch_size = tensorflow.shape(distribution_variance)[0]
             random = tensorflow.keras.backend.random_normal(shape=(batch_size,
         tensorflow.shape(distribution_variance)[1]))
             return distribution_mean + tensorflow.exp(0.5 * distribution_variance) *
         random

         distribution_mean = tensorflow.keras.layers.Dense(2, name='mean')(encoder)
```

```
distribution_variance = tensorflow.keras.layers.Dense(2,
name='log_variance')(encoder)

latent_encoding =
tensorflow.keras.layers.Lambda(sample_latent_features)([distribution_mean,
distribution_variance])
```

هذه الميزات المخفية (المحسوبة من التوزيع المكتسب) تكمل بالفعل مشفر النموذج. يمكن الآن تعريف نموذج المشفر على النحو التالي:

```
In [5]:  encoder_model = tensorflow.keras.Model(input_data,
         latent_encoding)
         encoder_model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)            [(None, 28, 28, 1)]  0           []

 conv2d (Conv2D)                 (None, 24, 24, 64)   1664        ['input_1[0][0]']

 max_pooling2d (MaxPooling2D)    (None, 12, 12, 64)   0           ['conv2d[0][0]']

 conv2d_1 (Conv2D)               (None, 10, 10, 64)   36928       ['max_pooling2d[0][0]']

 max_pooling2d_1 (MaxPooling2D)  (None, 5, 5, 64)     0           ['conv2d_1[0][0]']

 conv2d_2 (Conv2D)               (None, 3, 3, 32)     18464       ['max_pooling2d_1[0][0]']

 max_pooling2d_2 (MaxPooling2D)  (None, 1, 1, 32)     0           ['conv2d_2[0][0]']

 flatten (Flatten)               (None, 32)           0           ['max_pooling2d_2[0][0]']

 dense (Dense)                   (None, 16)           528         ['flatten[0][0]']

 mean (Dense)                    (None, 2)            34          ['dense[0][0]']

 log_variance (Dense)            (None, 2)            34          ['dense[0][0]']

 lambda (Lambda)                 (None, 2)            0           ['mean[0][0]',
                                                                   'log_variance[0][0]']

==================================================================================================
Total params: 57,652
Trainable params: 57,652
Non-trainable params: 0
```

المشفر بسيط للغاية ويحتوي فقط على حوالي 57000 معامل قابلة للتدريب.

التوزيع الكامن وأخذ العينات

يأخذ جزء المشفر من النموذج صورة كمدخل ومتجه المشفر الكامن كإخراج ، والذي يتم أخذ عينات من التوزيع المكتسب لمجموعة بيانات الإدخال. تتمثل مهمة وحدة فك التشفير في أخذ هذا المتجه المضمن كمدخل وإعادة إنشاء الصورة الأصلية (أو صورة تنتمي إلى نفس فئة الصورة الأصلية). نظرًا لأن المتجه المضمن هو تمثيل موجز جداً للخصائص ، فإن مفكك الشفرة يتكون من عدة أزواج من طبقات Deconvolutional وطبقات upsampling. طبقة Deconvolutional تعكس أساسًا ما تفعله الطبقة الالتفافية. تُستخدم طبقات upsampling لاستعادة الدقة الأصلية للصورة. بهذه الطريقة ، يعيد بناء الصورة بالأبعاد الأصلية.

```
In [5]:  decoder_input = tensorflow.keras.layers.Input(shape=(2))
         decoder = tensorflow.keras.layers.Dense(64)(decoder_input)
         decoder = tensorflow.keras.layers.Reshape((1, 1, 64))(decoder)
         decoder = tensorflow.keras.layers.Conv2DTranspose(64, (3,3),
         activation='relu')(decoder)

         decoder = tensorflow.keras.layers.Conv2DTranspose(64, (3,3),
         activation='relu')(decoder)
         decoder = tensorflow.keras.layers.UpSampling2D((2,2))(decoder)

         decoder = tensorflow.keras.layers.Conv2DTranspose(64, (3,3),
         activation='relu')(decoder)
         decoder = tensorflow.keras.layers.UpSampling2D((2,2))(decoder)

         decoder_output = tensorflow.keras.layers.Conv2DTranspose(1, (5,5),
         activation='relu')(decoder)
```

يمكن تعريف نموذج فك التشفير على النحو التالي:

```
In [5]:  decoder_model = tensorflow.keras.Model(decoder_input,
         decoder_output)
         decoder_model.summary()
```

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 2)]               0

 dense_1 (Dense)             (None, 64)                192

 reshape (Reshape)           (None, 1, 1, 64)          0

 conv2d_transpose (Conv2DTra  (None, 3, 3, 64)         36928
 nspose)

 conv2d_transpose_1 (Conv2DT  (None, 5, 5, 64)         36928
 ranspose)

 up_sampling2d (UpSampling2D  (None, 10, 10, 64)       0
 )

 conv2d_transpose_2 (Conv2DT  (None, 12, 12, 64)       36928
 ranspose)

 up_sampling2d_1 (UpSampling  (None, 24, 24, 64)       0
 2D)

 conv2d_transpose_3 (Conv2DT  (None, 28, 28, 1)        1601
 ranspose)

=================================================================
Total params: 112,577
Trainable params: 112,577
Non-trainable params: 0
```

بناء المشفر التلقائي القابل للتغيير

أخيرًا ، يمكن تعريف المشفر التلقائي القابل للتغيير من خلال الجمع بين أقسام التشفير وفك التشفير.

```
In [5]:  encoded = encoder_model(input_data)
         decoded = decoder_model(encoded)

         autoencoder = tensorflow.keras.models.Model(input_data, decoded)

         autoencoder.summary()
```

```
Model: "model_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 28, 28, 1)]       0

 model (Functional)          (None, 2)                 57652

 model_1 (Functional)        (None, 28, 28, 1)         112577

=================================================================
Total params: 170,229
Trainable params: 170,229
Non-trainable params: 0
_________________________________________________________________
```

دالة الخسارة

In [5]:
```
def get_loss(distribution_mean, distribution_variance):

    def get_reconstruction_loss(y_true, y_pred):
        reconstruction_loss = tensorflow.keras.losses.mse(y_true, y_pred)
        reconstruction_loss_batch = tensorflow.reduce_mean(reconstruction_loss)
        return reconstruction_loss_batch*28*28

    def get_kl_loss(distribution_mean, distribution_variance):
        kl_loss = 1 + distribution_variance - 
tensorflow.square(distribution_mean) - 
tensorflow.exp(distribution_variance)
        kl_loss_batch = tensorflow.reduce_mean(kl_loss)
        return kl_loss_batch*(-0.5)

    def total_loss(y_true, y_pred):
        reconstruction_loss_batch = get_reconstruction_loss(y_true, y_pred)
        kl_loss_batch = get_kl_loss(distribution_mean, distribution_variance)
        return reconstruction_loss_batch + kl_loss_batch

    return total_loss
```

أخيرًا ، النموذج جاهز للتدريب:

In [5]:
```
autoencoder.compile(loss=get_loss(distribution_mean, 
distribution_variance), optimizer='adam')
```

تدريب النموذج

In [5]:
```
autoencoder.fit(train_data, train_data, epochs=20, batch_size=64, 
validation_data=(test_data, test_data))
```

```
Epoch 7/20
60000/60000 [==============================] - 21s 342us/sample - loss: 34.7669 - val_loss: 34.9932
Epoch 8/20
60000/60000 [==============================] - 20s 340us/sample - loss: 34.5290 - val_loss: 34.4761
Epoch 9/20
60000/60000 [==============================] - 21s 342us/sample - loss: 34.3419 - val_loss: 34.1162
Epoch 10/20
60000/60000 [==============================] - 21s 344us/sample - loss: 34.1046 - val_loss: 33.9851
Epoch 11/20
60000/60000 [==============================] - 21s 342us/sample - loss: 33.9861 - val_loss: 33.8626
Epoch 12/20
60000/60000 [==============================] - 20s 341us/sample - loss: 33.8211 - val_loss: 33.9228
Epoch 13/20
60000/60000 [==============================] - 21s 343us/sample - loss: 33.6959 - val_loss: 34.0816
Epoch 14/20
60000/60000 [==============================] - 21s 342us/sample - loss: 33.6299 - val_loss: 33.8098
Epoch 15/20
60000/60000 [==============================] - 20s 341us/sample - loss: 33.4760 - val_loss: 33.9145
Epoch 16/20
60000/60000 [==============================] - 20s 339us/sample - loss: 33.3355 - val_loss: 33.6437
Epoch 17/20
60000/60000 [==============================] - 21s 343us/sample - loss: 33.3159 - val_loss: 33.6046
Epoch 18/20
60000/60000 [==============================] - 21s 343us/sample - loss: 33.1844 - val_loss: 33.4338
Epoch 19/20
60000/60000 [==============================] - 20s 341us/sample - loss: 33.1549 - val_loss: 33.4451
Epoch 20/20
60000/60000 [==============================] - 20s 340us/sample - loss: 33.0717 - val_loss: 33.5015
```

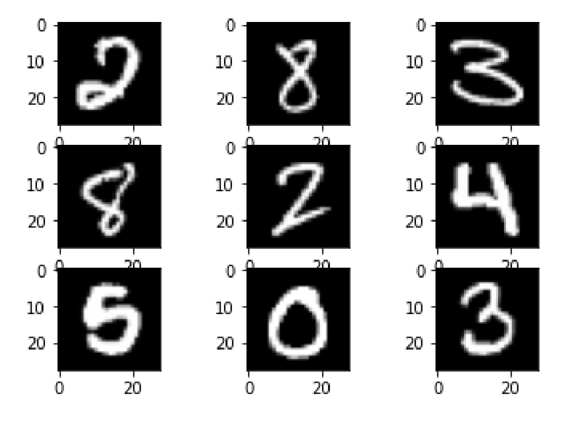
النتائج

في هذا القسم ، سنرى إمكانيات إعادة إنتاج نموذجنا على صور الاختبار. يختار الكود التالي 9 صور من مجموعة البيانات التجريبية ونرسم الصور المعاد بناؤها ذات الصلة بهم.

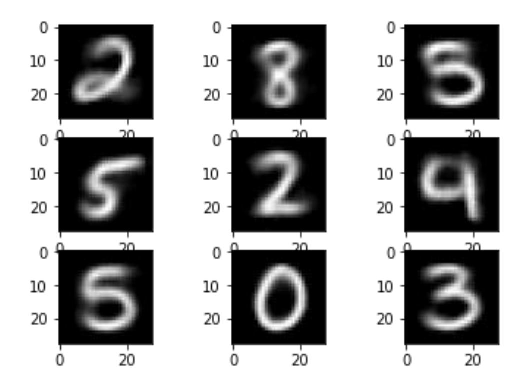
```
In [5]:     offset=400
            print ("Real Test Images")
            # Real Images
            for i in range(9):
                plt.subplot(330 + 1 + i)
                plt.imshow(test_data[i+offset,:,:, -1], cmap='gray')
            plt.show()

            # Reconstructed Images
            print ("Reconstructed Images with Variational Autoencoder")
            for i in range(9):
                plt.subplot(330 + 1 + i)
                output = autoencoder.predict(np.array([test_data[i+offset]]))
                op_image = np.reshape(output[0]*255, (28, 28))
                plt.imshow(op_image, cmap='gray')
            plt.show()
```

Real Test Images



Reconstructed Images with Variational Autoencoder



تظهر النتائج أعلاه أن النموذج قادر على إعادة بناء الصور الرقمية بالأداء المناسب. ومع ذلك ، فإن النقطة المهمة التي يجب ملاحظتها هنا هي أن بعض الصور المعاد بناؤها مختلفة جداً في المظهر عن الصور الأصلية. هذا لأن إعادة البناء لا تعتمد فقط على صورة الإدخال ، ولكن أيضًا على التوزيع الذي تم تعلمه. ممتع ، لا!

الشيء الثاني الذي يجب ملاحظته هنا هو أن الصور الناتجة ضبابية بعض الشيء. هذا أمر شائع في المشفرين التلقائيين القابلين للتغيير ، وغالبًا ما ينتجون مخرجات صاخبة (أو ذات جودة رديئة) لأن المتجهات الكامنة صغيرة جداً. المشفرات التلقائية المتغيرة ليست مصممة في الواقع لإعادة بناء الصور ، والغرض الحقيقي منها هو التعلم عن طريق التوزيع ، والذي يمنحها قوة هائلة لإنتاج بيانات مزيفة.

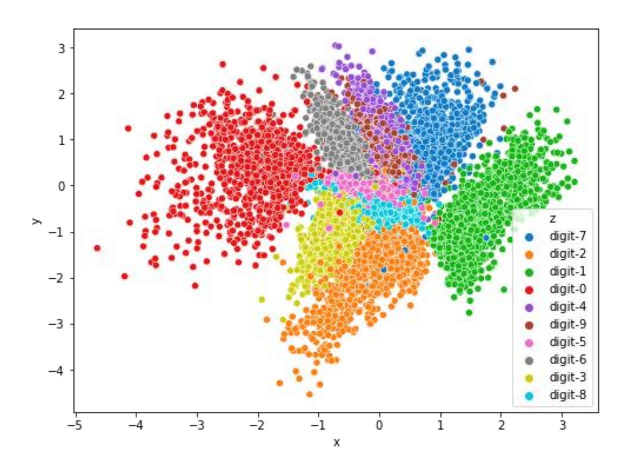
مجموعات الميزات الكامنة

كما ذكرنا سابقًا ، تتعلم المشفرات التلقائية المتغيرة التوزيع الأساسي للميزات الكامنة ، وهو ما يعني أساسًا أن الترميزات الكامنة للحالات التي تنتمي إلى فئة لا ينبغي أن تكون بعيدة جداً في المساحة الكامنة.

دعنا ننشئ تضمين كامن لجميع صورنا التجريبية ونمثلها مرئيا:

```
In  [5]:    x = []
            y = []
            z = []
            for i in range(10000):
                z.append(testy[i])
                op = encoder_model.predict(np.array([test_data[i]]))
                x.append(op[0][0])
                y.append(op[0][1])
            df = pd.DataFrame()
            df['x'] = x
            df['y'] = y
            df['z'] = ["digit-"+str(k) for k in z]

            plt.figure(figsize=(8, 6))
            sns.scatterplot(x='x', y='y', hue='z', data=df)
            plt.show()
```



يوضح الرسم البياني أعلاه أن تضمين نفس الأشكال في المساحة الكامنة يكون أقرب. هذا تقريبًا ما أردنا الحصول عليه من المشفرات التلقائية المتغيرة. لننتقل إلى النهاية ، حيث نختبر القدرات الإنتاجية لنموذجنا.

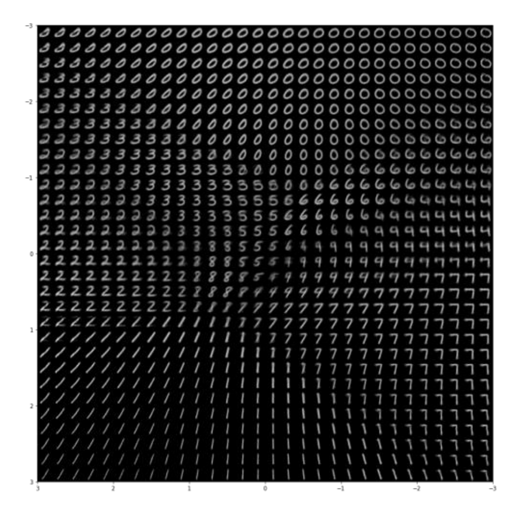
توليد الصور المزيفة

يوضح القسم السابق أن الترميزات الكامنة لبيانات الإدخال تتبع التوزيع الطبيعي القياسي والحدود الواضحة مرئية لفئات مختلفة من الأرقام. ومن ثم ، فإن النموذج يعرف أي جزء من الفضاء ينتمي إلى أي فئة. هذا يعني أنه يمكننا بالفعل إنتاج صور رقمية بخصائص مماثلة لمجموعة بيانات التدريب عن طريق تمرير نقاط عشوائية عبر الفضاء (مساحة التوزيع الكامنة). نتيجة لذلك ، يمكن استخدام المشفرات التلقائية المتغيرة كنماذج إنتاجية لإنشاء بيانات مزيفة.

دعونا ننشئ مجموعة من الأرقام باستخدام التشفير الكامن.

```
In  [5]:    generator_model = decoder_model
            x_values = np.linspace(-3, 3, 30)
            y_values = np.linspace(-3, 3, 30)
            figure = np.zeros((28 * 30, 28 * 30))
            for ix, x in enumerate(x_values):
                for iy, y in enumerate(y_values):
                    latent_point = np.array([[x, y]])
                    generated_image = generator_model.predict(latent_point)[0]
                    figure[ix*28:(ix+1)*28, iy*28:(iy+1)*28,] =
            generated_image[:,:,-1]

            plt.figure(figsize=(15, 15))
            plt.imshow(figure, cmap='gray', extent=[3,-3,3,-3])
            plt.show()
```



يمكنك العثور على جميع الأرقام (من 0 إلى 9) في مصفوفة الصورة أعلاه لأننا حاولنا إنتاج صور من جميع أجزاء المساحة الكامنة.

شبكة الخصومة التوليدية [1] (GANs)

تعد شبكات الخصومة التوليدية (GANs) واحدة من أقوى نماذج المولدات. هناك مكونان لبُنية شبكة GAN ، مسافات التمثيل D و G التي يجب أخذها في الاعتبار. يمكن فصل هذين المكونين عن طريق شبكتين عصبيتين. أحدهما هو الفاصل D والآخر هو المولد G. إنهم يتصرفون مثل لعبة ثنائية اللاعبين لتحسين بعضهم البعض خلال لعبة تنافسية. يحاول D التمييز بين الصور المزيفة التي تنتجها G من الصور الحقيقية ، بينما يحاول G إنتاج المزيد من الصور المشابهة للصور الحقيقية للتشويش على D. أخيرًا ، ينتج G المدرب صورًا واقعية.

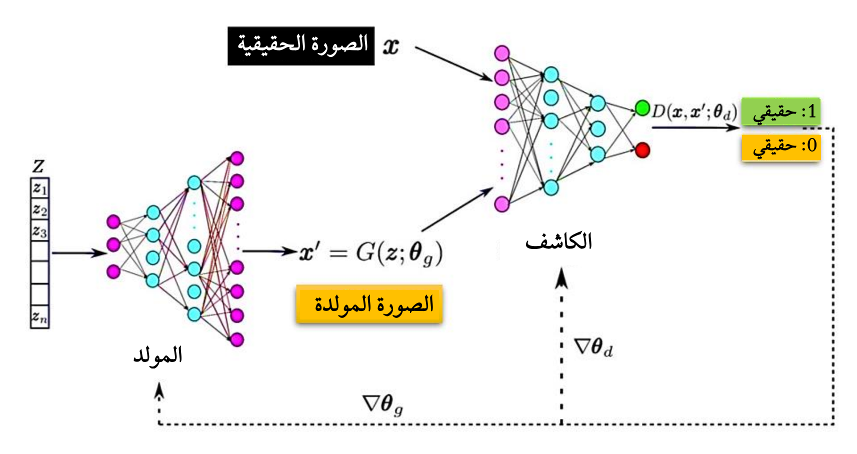
مدخلات المولد عبارة عن بعض الضوضاء العشوائية المأخوذة من الفضاء الكامن ، ومخرجاتها عبارة عن صورة x من المفترض أن تقدر الصور في مساحة البيانات الأصلية. افترض أن z هو متغير عشوائي مأخوذ من $p_z(z)$ و θ_g من معاملات التوليد G. وفقًا لذلك ، يمكن تمثيل خروج المولد على أنه $G(z;\theta_g)$. يبذل المولد G قصارى جهده لإنتاج صورة حقيقية بحيث يمكنه خداع الفاصل D. في هذه الأثناء ، يعمل الكاشف (الفاصل) D كمصنف ثنائي يأخذ الصور الأصلية (x) ومخرج $G(z;\theta_g)$ كمدخلات. بعد ذلك ، يحاول الفاصل D المزود بالمعاملات θ_d التمييز بين الصور الأصلية والصورة المزيفة. الهدف من شبكة الخصومة التوليدية هو رياضيا على النحو التالي:

[1] Generative adversarial networks

$$\min_G max_D L(D,G) = \mathbb{E}_{x\sim p_r(x)} log D(x,\theta_d) + \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z;\theta_z)))$$

في المعادلة أعلاه ، $p_r(x)$ هو توزيع البيانات الحقيقية و $p_g(x)$ هو توزيع البيانات التي تم إنشاؤها بواسطة المولد.

يمكن للكاشف D التعرف بسهولة على الصور الحقيقية والمزيفة في المراحل الأولى من التدريب. ومن ثم ، فإن $\log(1 - D(G(z;\theta_z))$ مشبع. عمليًا ، يمكننا استبدال $\log\ G(z;\theta_z)$ بدلاً من تصغير $\log(1 - D(G(z;\theta_z))$ يظهر إطار شبكات الخصومة التوليدية في الشكل 8ـ 24.



الشكل 8ـ24. شبكة الخصومة التوليدية.

خلاصة الفصل الثامن

- يتألق التعلم غير الخاضع للإشراف للمشكلات التي تكون فيها الأنماط غير معروفة ، أو تتغير باستمرار ، أو التي ليس لدينا مجموعات بيانات مصنفة كافية لها.
- يجعل التعلم غير الإشرافي المشكلات غير القابلة للحل سابقًا أكثر قابلية للحل وأكثر مرونة في العثور على أنماط مخفية في كل من البيانات السابقة المتاحة للتدريب والبيانات المستقبلية.
- مع التعلم غير الخاضع للإشراف ، يمكن تسمية العينات غير المسماة تلقائيًا.
- المجموعات هي المناطق التي تكون فيها كثافة نقاط البيانات المتشابهة عالية.
- التجميع (الكلاستيرينك) هو مهمة تجميع مجموعة من الكائنات بحيث تكون الكائنات في مجموعة (تسمى الكلاستر او العنقود) أكثر تشابهًا مع بعضها البعض من المجموعات الأخرى (العناقيد).

- يمكن استخدام التجميع بمفرده لتحديد البنية الجوهرية للبيانات ، في حين أنه يمكن أيضًا أن يكون بمثابة تقنية معالجة مسبقة لمهام التعلم الأخرى مثل التصنيف.
- الغرض من التجميع وصفي والغرض من التصنيف تنبئي.
- تستخدم العديد من طرق التجميع معايير المسافة لتحديد التشابه أو الاختلاف بين كل زوج من الكائنات.
- تجميع النموذج الأولي هو عائلة من خوارزميات التجميع التي تفترض أن بنية التجميع يمكن تمثيلها بواسطة مجموعة من النماذج الأولية.
- تتبنى خوارزمية k-mean استراتيجية جشعة وتعتمد طريقة تحسين تكرارية للعثور على الحل التقريبي.
- تتمثل إحدى نقاط القوة في تجميع نموذج غاوسي المختلط في أنه طريقة تجميع ناعمة.
- النموذج المختلط غاوسي هو ببساطة نموذج يناسب توزيعات غاوسيان المتعددة في مجموعة من البيانات.
- يمثل كل غاوسي في النموذج المختلط مجموعة محتملة.
- المجموعات الهرمية لها نهج مختلف ، وكما يوحي اسمها ، فإن التسلسل الهرمي يطور مجموعات في شكل أشجار.
- نهج التجميع القائم على الكثافة هو طريقة قادرة على إيجاد مجموعات من الشكل المطلوب ، وكما يوحي اسمها ، تستخدم كثافة العينة لتعيين عضوية المجموعة.
- تقوم خوارزميات التجميع المستندة إلى الكثافة بتقييم العلاقة بين العينات من منظور الكثافة وتوسيع المجموعات عن طريق إضافة عينات ذات صلة.
- على عكس العديد من خوارزميات التجميع التقليدية الأخرى ، فإن خوارزميات التجميع القائمة على الكثافة لديها القدرة على التعامل مع الحالات البعيدة.
- DBSCAN هي خوارزمية تجميع تعتمد على الكثافة تحدد كثافة توزيعات العينة مع زوج من معاملات "الجوار" ε, μ.
- ستحصل على أقصى استفادة من التجميع عندما تستخدم التجميع ليس كنموذج قائم بذاته ، ولكن كجزء من إستراتيجية أوسع لاكتشاف البيانات.
- يعد تجنب الضبط الزائد هو الدافع الرئيسي لتقليل الأبعاد.
- تعني الأبعاد الأقل في البيانات وقت تدريب أقل وموارد حسابية أقل.
- تقليل الأبعاد مفيدة جدًا للتمثيل المرئي للبيانات.
- يزيل تقليل الابعاد الضوضاء في البيانات.
- يحاول نهج اختيار الميزة تحديد مجموعة فرعية من الميزات المهمة وإزالة الميزات غير المهمة.
- يحاول اختيار الميزة إنشاء مساحة فرعية جديدة للسمة.

- الفكرة الرئيسية وراء استخراج الميزات هي ضغط البيانات بهدف الحفاظ على المزيد من المعلومات ذات الصلة.
- المشفرات التلقائية عبارة عن شبكات عصبية تستخدم لتقليل الأبعاد.
- تتكون المشفرات التلقائية من جزأين من الشبكة العصبية ، المشفر وجهاز فك التشفير.
- النماذج الإنتاجية (التوليدية) هي فئة من نماذج التعلم الآلي المستخدمة لوصف كيفية إنشاء البيانات.
- الغرض من النموذج التوليدي هو الوصف الكامل لمجموعة البيانات الممكنة.

مصادر إضافية لمزيد من القراءة

Albalate, A., & Minker, W. (2013). Semi-supervised and unsupervised machine learning: novel strategies. John Wiley & Sons.

Rhys, H. (2020). Machine Learning with R, the tidyverse, and mlr. Simon and Schuster.

Tripathy, B. K., Anveshrithaa, S., & Ghela, S. (2021). Unsupervised Learning Approaches for Dimensionality Reduction and Data Visualization: Unsupervised Learning Approaches for Dimensionality Reduction and Data Visualization. CRC Press.

Zhao, H., Lai, Z., Leung, H., & Zhang, X. (2020). Feature Learning and Understanding: Algorithms and Applications. Springer Nature.

9

موضوعات مختارة

الأهداف:

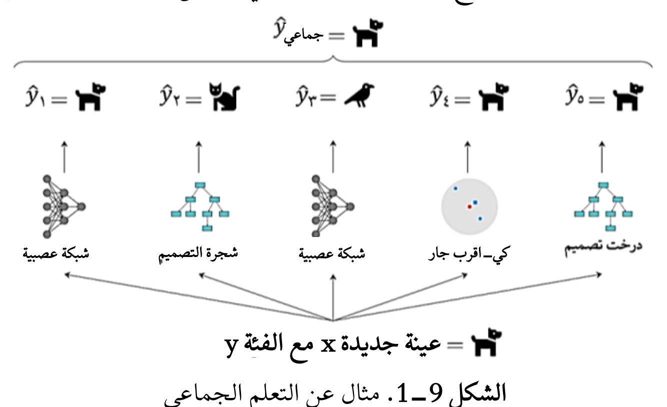
- ما هو التعلم الجماعي وما هي فوائده؟
- التعرف على التعلم مدى الحياة وعلاقته بالطرق الأخرى.
- ما هو التعلم المعزز؟
- خوارزميات التعلم المعزز.
- التعلم بالمحاكاة واختلافه عن التعلم المعزز.

التعلم الجماعي[1]

في كثير من الحالات عندما نحتاج إلى اتخاذ قرارات مهمة ، فمن طبيعتنا طلب مساعدة خبير للمساعدة في اتخاذ القرار. في معظم الحالات ، نتقدم خطوة أخرى ونبحث عن وجهات النظر الثانية والثالثة أيضًا. هذا لأننا نعتقد أنه لا يمكن لأحد بمفرده أن يكون لديه معرفة كاملة بموضوع صعب من الناحية الموضوعية (على سبيل المثال ،التشخيص الطبي). بالنظر إلى عدد من العوامل، فإننا نزن ونجمع بشكل بديهي التوصيات التي نتلقاها لمعرفة قرارنا النهائي. نستخدم كل هذه القرارات لأننا نتوقع منها أن تكون أفضل من القرارات التي نتخذها.

على الرغم من مهاراتنا في حل المشكلات ، لا يزال البشر يستفيدون من استشارة مصادر متعددة. لذلك ، من الطبيعي أن يكون مجال التعلم الآلي مستوحى من هذه العادة البشرية ، عندما يبحثون عن طرق لتحسين نماذجهم. يُعرف مجال التعلم الآلي الذي يتضمن إجراءات لتحسين أداء النموذج من خلال تدريب نماذج وطرق متعددة للجمع بين مخرجاتها باسم **التعلم الجماعي**. بعبارات أبسط ، التعلم الجماعي هو فن استخدام نماذج متعددة لتحقيق أداء تنبؤي أفضل.

الفرضية الأساسية وراء النماذج الجماعية هي أن مجموعة من المتعلمين الضعفاء يجتمعون لمساعدة بعضهم البعض في تكوين متعلم قوي. الحدس الكامن وراء النمذجة الجماعية مرادف لما اعتدنا عليه في حياتنا اليومية ، مثل طلب المشورة من العديد من الخبراء قبل اتخاذ قرار معين لتقليل احتمالية اتخاذ قرار سيء أو نتيجة عكسية. يظهر السيناريو الذي يتم فيه تكوين مجموعة مفيدة في الشكل 9ـ1. في هذا المثال ، نقوم بتدريب 5 نماذج على نفس بيانات التدريب ، كل نموذج يصوت لفئة الصورة ، وتتوقع المجموعة الفئة الذي حصل على أكبر عدد من الأصوات.



الشكل 9ـ1. مثال عن التعلم الجماعي

لا يوجد نموذج للتعلم الآلي مثالي. جميعهم لديهم قيود ويرتكبون أخطاء ؛ مثلما قد يطلب الإنسان مشورة العديد من الخبراء قبل اتخاذ قرار صعب. وقد ثبت أيضًا أن مجموعة التنبؤات لعدة نماذج توفر تنبؤات أكثر دقة. حددت العديد من الدراسات النظرية والتجريبية المواقف التي

[1] Ensemble learning

تكون فيها أنظمة المجموعة مفيدة. بشكل عام ، هناك ثلاثة أسباب تجعل التعلم الجماعي ينتج نتائج أفضل من المتعلم الفردي:

1. **بيانات تدريبية محدودة:** بالنسبة للمشكلات المعقدة ، قد لا توفر البيانات معلومات كافية لجميع خصائص البيانات المحتملة التي قد يواجهها المتعلم. وبالتالي، من غير المرجح أن تعمل فرضية واحدة لمجموعة بيانات محدودة بسبب عدم كفاية المعلومات لمجموعة بيانات أخرى. في البيانات التدريبية ، من خلال الجمع بين فرضيات المتعلمين ، هناك فرصة أكبر في أن يتمكن المتعلم الجماعي من إدارة البيانات ذات الخصائص غير المعروفة بنجاح.
2. **التعويض عن عمليات البحث غير المكتملة:** قد تكون عمليات البحث الخاصة بخوارزميات التعلم غير مكتملة. وبالتالي ، حتى لو كانت هناك فرضية مثالية فريدة ، فقد لا يتم العثور عليها أبدًا. يمكن أن يساعد التعلم الجماعي في تعويض عمليات البحث المعيبة هذه.
3. **لا توجد فرضية واحدة تناسب الجميع:** قد لا تحتوي مساحة البحث لمجموعة بيانات معينة على فرضية مثالية واحدة. ومن ثم يمكن أن يوفر التعلم الجماعي تقديرات تقريبية جيدة من خلال الجمع بين فرضيات متعددة.

هناك التعلم العميق، لماذا التعلم الجماعي؟!

أدت التطورات الحسابية والمنهجية الحديثة فيما يعرف بالتعلم العميق إلى تغيير الفائدة المتصورة لأساليب التعلم الجماعي. من خلال البنية الملائمة وتعديل المعاملات، يمكن للشبكات العصبية العميقة أداء وظيفة شبه مثاليةفي مجموعة واسعة من مجموعات البيانات. بالإضافة إلى ذلك، تتوفر العديد من التدابير التنظيمية لضمان تقارب هذه النماذج المعقدة دون حدوث الضبط الزائد.

إذن مع هذا الأداء المذهل للتعلم العميق، هل ما زالت الأساليب الجماعية تلعب دورًا مفيداً؟! أولاً، التدريب على الشبكات العصبية العميقة له تكلفة حسابية عالية. نادرًا ما يكون الإعداد الشامل للمعاملات لهذه النماذج ممكنًا بسبب نقص الموارد الحسابية. ومع ذلك، فإن العثور على البنية المناسبة لإنتاج نماذج ذات أداء تنبؤي عالٍ أمر ضروري.في المقابل، يمكن لنظام المجموعة أن ينتج نفس أداء التنبؤ مثل هذه النماذج الفردية، ولكن دون تكبد تكاليف حسابية مماثلة. قد يبدو الأمر بديهيًا للوهلة الأولى، لأن العديد من النماذج تحتاج أيضًا إلى التدريب لإنتاج مجموعة! ومع ذلك، يمكن أن تكون النماذج الفردية أقل تعقيداً (طبقات أقل)، مما يسمح باستكشاف أفضل لمساحة المصنففي نفس الميزانية الحسابية. بالإضافة إلى ذلك، يمكن اعتبار العديد من هذه الإعداداتفي نظام المجموعة النهائي (بدلاً من اختيار أفضل نموذج).

ثانيًا، لا تمنع الأساليب التنظيمية تمامًا الضبط الزائد. لذلك، فإن الجمع بين النماذج التي لم تتقارب، أو التي تعاني من مشكلة الضبط الزائد للبيانات، يمكن أن ينتج نماذج أفضل دون إعادة التدريب. أخيرًا، عند استكشاف مجموعات البياناتفي العالم الحقيقي، يُلاحظفي العديد من المسابقات أن أنظمة المجموعة تنتج أفضل النتائج.

بينما حققت الشبكات العصبية العميقة أداءً قريبًا من البشر في بعض المهام ، فإن الحقيقة هي أن هذه النماذج عالية التخصص. ومن ثم ، فإن استخدام الشبكة العصبية التي يمكن أن تكون دقيقة للغاية عند التمييز بين صور القطط والكلاب وتطبيقها على مشكلة مختلفة ولكنها ذات صلة (على سبيل المثال ، تحديد الفئات المتشابهة بناءً على عينات الفيديو) يؤدي إلى أداء ضعيف للغاية. لا يتعلق الأمر بموضوع شائع جدًا ، قضية الغداء المجاني (تمت مناقشته سابقًا): **لن يعمل نموذج واحد للتعلم الآلي بشكل أفضل على جميع أنواع البيانات ومجموعات البيانات.** هذا يدل على أن مجموعة من النماذج عالية التخصص هي أفضل حل لهذه المشكلة.

تقنيات التعلم الجماعي

التعلم الجماعي هو نموذج للتعلم الآلي يتم فيه تدريب العديد من النماذج ، والتي يشار إليها غالبًا باسم "**المتعلمين الفقراء**" ، على حل مشكلة والجمع لتحقيق نتائج أفضل. الفرضية الرئيسية هي أنه عندما يتم الجمع بين النماذج الضعيفة بشكل صحيح ، يمكننا الحصول على نماذج أكثر دقة / أقوى.

في نظرية التعلم الجماعي ، نسمي نماذج **المتعلمين الضعفاء** (أو النماذج الأساسية) التي يمكن استخدامها ككتل لتصميم نماذج أكثر تعقيدًا من خلال الجمع بين عدة نماذج. في كثير من الأحيان ، لا تعمل هذه النماذج الأساسية بشكل جيد بمفردها نظرًا لارتفاع درجة تحيزها أو تباينها. ومن ثم ، فإن فكرة الأساليب الجماعية هي محاولة تقليل التحيز أو التباين لدى هؤلاء المتعلمين الضعفاء من خلال الجمع بين العديد منهم لإنشاء **متعلم قوي** (أو نموذج جماعي) يحقق أداءً أفضل.

لإنشاء نموذج تعليمي جماعي ، يجب علينا أولاً تحديد نماذجنا الأساسية للتجميع. في أغلب الأحيان ، يتم استخدام خوارزمية التعلم الأساسية للحصول على متعلمين ضعفاء متجانسين تم تدريبهم بطرق مختلفة. تسمى هذه الأنواع من النماذج "**متجانسة**". ومع ذلك ، هناك طرق تستخدم أنواعًا مختلفة من خوارزميات التعلم الأساسية. يتم بعد ذلك دمج بعض المتعلمين غير المتجانسين بشكل ضعيف في "نموذج مجموعة غير متجانس".

نقطة مهمة هي أن اختيار المتعلمين الضعفاء يجب أن يكون متسقًا مع طريقة تجميع هذه النماذج. إذا اخترنا النماذج الأساسية ذات التحيز المنخفض ولكن التباين العالي ، فيجب أن يكون ذلك مع طريقة التجميع التي تميل إلى تقليل التباين ، بينما إذا اخترنا النماذج الأساسية

ذات التباين المنخفض ولكن التحيز العالي ، فيجب أن تكون بالطريقة التراكمية التي تميل إلى تقليل له تحيز.

هناك العديد من التقنيات الجماعية للجمع بين متعلمي الآلة المتعددين لإنشاء نموذج تنبؤي. الأساليب الأكثر شيوعًا هي: **التجميع**[1] لتقليل التباين **التعزيز**[2] لتقليل التحيز.

التعزيز (بوستينك)

التعزيز عبارة عن مجموعة من الخوارزميات التي تحول المتعلمين الضعفاء إلى متعلمين أقوياء. تبدأ خوارزمية التعزيز بتدريب متعلم أساسي واحد ثم تعدل توزيع عينات التدريب وفقًا لنتيجة المتعلم الأساسي بحيث تكون عينات التصنيف غير الصحيحة أكثر وضوحًا للمتعلمين الأساسيين التاليين. بعد تدريب المتعلم الأساسي الأول ، يتم تدريب المتعلم الأساسي الثاني على عينات تدريب معدلة ويتم استخدام النتيجة لإعادة ضبط توزيع عينة التدريب. تتكرر هذه العملية حتى يصل عدد المتعلمين الأساسيين إلى القيمة المحددة مسبقًا T و في نهاية المطاف يصبح هؤلاء المتعلمون أساس الوزن والتعزيز.

خوارزمية التعزيز الأكثر شهرة هي **AdaBoost** ، والتي تعمل على تحسين خوارزمية التعزيز البسيطة من خلال عملية تكرارية. الفكرة الرئيسية وراء هذه الخوارزمية هي التركيز أكثر على الأنماط التي يصعب تصنيفها. يتم تحديد مقدار التركيز من خلال الوزن المخصص لكل نمط في مجموعة التدريب. في البداية ، يتم تعيين نفس الوزن لجميع الأنماط. مع كل تكرار ، يزداد وزن جميع عينات التصنيف غير الصحيحة بينما ينخفض وزن عينات التصنيف الصحيحة. نتيجة لذلك ، يضطر المتعلم الفقير إلى التركيز على الأمثلة الصعبة للمجموعة التدريبية عن طريق إجراء عمليات تكرار إضافية وإنشاء المزيد من الفئات. بالإضافة إلى ذلك ، يتم تعيين الوزن لكل فئة. يقيس هذا الوزن الدقة الإجمالية للتصنيف وهو دالة على الوزن الإجمالي لأنماط التصنيف الصحيحة. لذلك ، يتم إعطاء أوزان أعلى للفئات الأكثر دقة. تستخدم هذه الأوزان لتصنيف الأنماط الجديدة.

من منظور تحليل التباين التحيز ، يركز التعزيز بشكل أساسي على تقليل التحيز. هذا هو السبب في أن مجموعة من المتعلمين ذوي القدرة التعميمية الضعيفة يمكن أن تكون قوية للغاية.

---

[1] bagging

[2] boosting

التجميع (باكينيك)

التجميع هو طريقة بسيطة لكنها فعالة لإنشاء مجموعة من الفئات. يدمج مصنف المجموعة الذي تم إنشاؤه بهذه الطريقة مخرجات المصنفات المختلفة في مصنف واحد. ينتج عن هذا فئة أكثر دقة من أي فئة أخرى.

الفكرة من وراء التجميع هي الجمع بين نتائج عدة نماذج للحصول على نتيجة إجمالية. هناك سؤال هنا: هل سيكون مفيداً إذا تمكنا من بناء جميع النماذج على مجموعة بيانات واحدة ودمجها؟ من المحتمل جداً أن يكون لهذه النماذج نفس النتيجة ، لأن لها نفس المدخلات. فكيف يمكننا حل هذه المشكلة؟ تتمثل إحدى الطرق في إنشاء متعلمين أساسيين مختلفين عن طريق تقسيم مجموعة التدريب الرئيسية إلى عدة مجموعات فرعية غير متداخلة واستخدام كل مجموعة فرعية لتدريب المتعلم الأساسي. نظرًا لاختلاف مجموعات التدريب الفرعية ، فمن المرجح أن يكون المتعلمون الأساسيون المدربون مختلفين أيضًا. ومع ذلك ، إذا كانت المجموعات الفرعية مختلفة تمامًا ، فهذا يعني أن كل مجموعة فرعية تحتوي فقط على جزء صغير من مجموعة التدريب الرئيسية ، والتي من المحتمل أن تؤدي إلى ضعف التعلم. نظرًا لأن المجموعة الجيدة تتطلب أن يكون كل متعلم أساسي جيداً بشكل معقول ، فإننا غالبًا نسمح للمجموعات الفرعية بالتداخل بطريقة تحتوي كل منها على عينات كافية.

التجميع يعمل على أساس أخذ عينات **Bootstrap**. بالنظر إلى مجموعة بيانات مع عينات m، فإنه يختار عينة بشكل عشوائي ونسخها في مجموعة العينات. بعد ذلك ، نحتفظ به في مجموعة البيانات الرئيسية حتى تظل هناك فرصة لإزالته في المرة القادمة. بتكرار هذه العملية m مرات ، يتم الحصول على مجموعة بيانات تحتوي على عينات m والتي قد تظهر فيها بعض العينات الأصلية أكثر من مرة بينما قد لا يظهر بعضها أبداً. نعلم من الفصل الخامس أن ما يقرب من 63.2٪ من العينات الأصلية تظهر في مجموعة البيانات.

يؤدي تطبيق العملية المذكورة أعلاه بالترتيب T ، إلى إنتاج مجموعة بيانات T ، كل منها يحتوي على عينات m. ثم يتم تدريب المتعلمين الأساسيين ودمجهم في مجموعة البيانات هذه. مثل هذا الإجراء هو عملية التجميع الأولية. عند الجمع بين تنبؤات المتعلمين الأساسية ، يعتمد التجميع طريقة التصويت البسيطة للمهام الفئوية وطريقة حساب المعدل البسيط لمهام الانحدار. عندما تحصل فئات متعددة على نفس عدد الأصوات ، يمكننا اختيار واحد أو أكثر بشكل عشوائي لاختبار موثوقية الأصوات.

يجلب أخذ عينات Bootstrap ميزة أخرى للتجميع: نظرًا لأن كل متعلم أساسي يستخدم فقط 63.2٪ من العينات التدريبية الأصلية للتدريب ، يمكن استخدام 36.8٪ من العينات المتبقية (خارج الحقيبة) كمجموعة تحقق من القدرة على التعميم. للحصول على هذا التقدير ، نحتاج إلى تتبع الأنماط التعليمية المستخدمة من قبل كل متعلم أساسي. لنفترض أن D_t تمثل مجموعة

من الأمثلة التي يستخدمها المتعلم h_t و $H^{oob}(x)$ تمثل تنبوًأ للعينة x لم تتم رؤيتها ، أي ، ضع في اعتبارك فقط التنبؤات التي قدمها المتعلمون الأساسيون الذين لم يستخدموا عينة x للتدريب. ومن ثم لدينا:

$$H^{oob}(x) = arg_{y\epsilon Y}\ max \sum_{t=1}^{T} \mathbb{I}(\ h_t(x) = y). \mathbb{I}(x \notin D_t)$$

وتقدير العينات غير المرئية هو خطأ تعميم للتجميع:

$$\epsilon^{oob}(x) = \frac{1}{|D|} \sum_{(x.y)\epsilon D} \mathbb{I}(H^{oob}(x) \neq y)$$

افترض أن التعقيد الحسابي هو متعلم أساسي $O(m)$ ، وبالتالي فإن تعقيد التجميع هو تقريبًا $T(O(m) + O(s))$ هو تعقيد التصويت أو المتوسط. نظرًا لأن تعقيد $O(s)$ منخفض و T ثابت غالبًا ما لا يكون كبيرًا جدًا ، فإن التجميع له نفس التعقيد الأساسي للمتعلم ، مما يعني أن التجميع هو خوارزمية تعلم جماعي فعالة.

من وجهة نظر تحليل التباين_التحيز ، يساعد التجميع على تقليل التباين ، وهذا مفيد بشكل خاص لأشجار القرار غير المعالجة.

الفرق بين التجميع والتعزيز

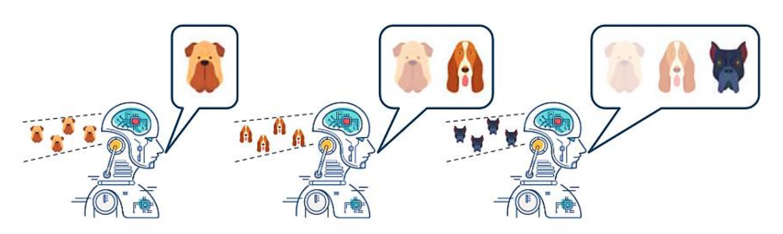
التجميع ، مثل التعزيز ، هو تقنية تعمل على تحسين دقة التصنيف من خلال إنتاج نموذج هجين يجمع بين تصنيفات متعددة. تتبع كلتا الطريقتين نهج التصويت الذي يتم إجراؤه بشكل مختلف للجمع بين مخرجات الفئات المختلفة. في التعزيز ، على عكس التجميع ، تتأثر كل فئة بأداء الفئات التي تم إنشاؤها قبل إنشائها. على وجه الخصوص ، يولي المصنف الجديد مزيدًا من الاهتمام لأخطاء التصنيف التي ارتكبتها المصنفات السابقة ، حيث يتم تحديد مقدار الاهتمام من خلال أدائها. في التجميع ، يتم اختيار كل عينة باحتمالية متساوية ، بينما في التعزيز ، يتم اختيار العينات مع احتمال يتناسب مع وزنها.

التعلم مدى الحياة[1]

مع توفر مجموعات بيانات أكبر وخفض التكاليف الحسابية ، أصبحت النماذج القادرة على حل المهام الأكبر متاحة. ومع ذلك ، قد يكون تدريب نموذج في كل مرة تحتاج فيها إلى تعلم مهمة جديدة أمرًا مستحيلًا. نظرًا لأن البيانات القديمة قد لا تكون متاحة ، فقد لا يتم تخزين البيانات الجديدة بسبب مشكلات الخصوصية ، أو قد لا يدعم تكرار تحديث النظام تدريب نموذج جديد مع تكرار جميع البيانات بشكل كافٍ. يمكن العثور على حلول لهذه المشاكل في **التعلم مدى**

[1] lifelong

الحياة. الهدف من التعلم مدى الحياة هو أن تكون قادرًا على تعلم مهام جديدة دون الحاجة إلى الوصول إلى البيانات الخاصة بالمهام التي تعلمناها بالفعل. عندما تتعلم الشبكات العصبية مهامًا جديدة ، فإن المعرفة الجديدة لها الأسبقية على المعرفة القديمة إذا لم يتم استخدام معايير معينة، مما يؤدي غالبًا إلى نسيان المعرفة الثانوية. هذا هو المعروف باسم **النسيان الكارثي**[1] (انظر الشكل 9ــ2). يحدث النسيان الكارثي عندما تكون الشبكة العصبية المدربة غير قادرة على الحفاظ على قدرتها على أداء المهام التي تعلمتها بالفعل ، عندما يتم تكييفها لأداء مهام جديدة.



شكل 9ــ2. صورة النسيان الكارثي. تُنسى المعرفة التي تم تعلمها سابقًا (تختفي تدريجياً) عند تعلم فئات جديدة لم تتم رؤيتها لفترة من الوقت.

التعلم مدى الحياة هو نموذج متقدم للتعلم الآلي يتعلم باستمرار ، ويجمع المعرفة المكتسبة في العمل السابق ، ويستخدمها للمساعدة في التعلم في المستقبل. في هذه العملية ، يصبح المتعلم أكثر وعيًا وفعالية في التعلم. القدرة على التعلم هي إحدى خصائص الذكاء البشري. ومع ذلك ، يتم تعلم نمط التعلم الآلي السائد الحالي بشكل منفصل: وفقًا لمجموعة بيانات التدريب، تعمل خوارزمية التعلم الآلي على مجموعة البيانات لإنشاء نموذج. لا يبذل أي جهد للحفاظ على المعرفة المكتسبة وتطبيقها على التعلم في المستقبل. على الرغم من أن نموذج التعلم المنفصل هذا كان ناجحًا للغاية ، إلا أنه يتطلب عددًا كبيرًا من أمثلة التدريب وهو مناسب فقط للمهام المحددة والمحدودة جيداً.

بالمقارنة ، يمكننا نحن البشر أن نتعلم بفعالية من خلال بعض الأمثلة. لأننا تراكمت لدينا في الماضي الكثير من المعرفة التي تتيح لنا التعلم بالبيانات أو بجهد ضئيل. الهدف من التعلم مدى الحياة هو تحقيق هذه القدرة. تتطلب برامج مثل المساعدين الأذكياء والروبوتات الحوارية والروبوتات المادية التي تتفاعل مع البشر والأنظمة في بيئات العالم الحقيقي أيضًا قدرات التعلم مدى الحياة هذه. بدون القدرة على جمع المعرفة المكتسبة واستخدامها للتعلم التدريجي ، ربما لن يكون النظام ذكيًا حقًا على الإطلاق.

[1] catastrophic forgetting

تعريف التعلم مدى الحياة

المتعلم الذى تعلم فى سلسلة من المهام، من 1 إلى $N-1$. عندما يواجه المهمة N، يستخدم المعرفة المكتسبة من مهام $N-1$ السابقة للمساعدة فى تعلم المهمة N.

للتأكيد على السياق الدائم، نسمى المهام 1 من خلال المهام / المجالات السابقة N-1 والمهام / المجالات الحالية لـ N.

هناك العديد من الأسئلة والتحديات في تصميم نظام التعلم الآلي مدى الحياة[1] (LML):

- **ما هي المعلومات التي يجب الاحتفاظ بها من مهام التعلم السابقة؟**
- **ما هي أشكال المعرفة التي سيتم استخدامها للمساعدة في التعلم في المستقبل؟**
- **كيف يكتسب النظام المعرفة؟**
- **كيف يستخدم النظام المعرفة للمساعدة في التعلم في المستقبل؟**

للإجابة على الأسئلة أعلاه، يحتاج نظام LML إلى المكونات العامة الأربعة التالية:

- **تخزين المعلومات السابقة** [2] **(PIS):** يخزن المعلومات من التعلم السابق. قد يشمل ذلك مستودعات فرعية للمعلومات مثل (1) البيانات الأصلية المستخدمة في كل مهمة سابقة ، (2) نتائج التعلم الوسيطة لكل مهمة سابقة ، و (3) النموذج النهائي أو الأنماط المستفادة من المهام السابقة.
- **قاعدة المعرفة** [3] **(KB):** تخزن المعرفة المستخرجة أو المدمجة من PIS. هذا يتطلب مخطط تمثيل المعرفة المناسب للتطبيق. تعد قابلية توسيع قاعدة المعرفة ضرورية أيضًا للبيانات الوصفية.
- **مستخرج المعرفة** [4] **(KM):** يستخرج المعرفة من PIS. يمكن اعتبار هذا التنقيب بمثابة عملية تعلم. لأنه يتعلم المعرفة من المعلومات المكتسبة من تعلم المهام السابقة. تتم إضافة المعرفة إلى قاعدة المعارف الموجودة (المعرفة الأساسية).
- **المتعلم القائم على المعرفة (KBL):** اعتمادًا على المعرفة المتوفرة في KB ، يمكن لهذا المتعلم استخدام المعرفة أو بعض المعلومات في PIS لوظيفة جديدة.

من منظور آخر، تعتبر العناصر التالية ضرورية لعامل LML:

(1) الحفاظ على المعرفة بالمهمة المكتسبة.

---

[1] Lifelong Machine Learning

[2] Past Information Store

[3] Knowledge Base

[4] Knowledge Miner

(2) النقل الانتقائي أو الاستخدام الانتقائي للمعرفة السابقة عند حل المهام الجديدة.
(3) نهج منهجي يضمن التفاعل الفعال والكفوء لعناصر الاحتفاظ والإرسال المذكورة أعلاه.

في هذا السياق ، تتم مناقشة الحفاظ على المعرفة من منظور تمثيل المعرفة. يمكن تمثيل كل متعلم بطرق مختلفة. يمكن أن يكون أبسط شكل هو تخزين عينات التدريب ببساطة. يتميز تخزين البيانات التدريبية الأولية بالدقة ونقاء المعرفة (الاحتفاظ بالمعرفة). ومع ذلك ، فهي غير فعالة بسبب كمية التخزين الكبيرة التي تتطلبها. بدلاً من ذلك ، يمكن تخزين تمثيل لفرضية دقيقة تم إنشاؤها من أمثلة التدريب. تتمثل مزايا المعرفة التمثيلية في صغر حجمها مقارنة بالمساحة المطلوبة للبيانات التدريبية الأصلية وقدرتها على التعميم بما يتجاوز الأمثلة التدريبية.

هناك حديث عن نقل المعرفة من منظور التعلم الآلي. يتضمن نقل التمثيل تعيين تمثيل مهمة معروفة لنظام التعلم بمهمة موضوعية جديدة. من خلال القيام بذلك ، يتم تحديد النموذج الجديد كمياً في منطقة معينة من مساحة فرضية النظام. غالبًا ما يقلل نقل التمثيل من وقت التدريب للنموذج الجديد دون المساس بشكل كبير بأداء تعميم الفرضيات الناتجة. يؤكد النهج المنهجي على التفاعل الأساسي بين الاحتفاظ بالمعرفة والتعلم الانتقالي. LML ليست مجرد خوارزمية. يمكن أن تستفيد LML من الأبحاث الجديدة حول خوارزميات التعلم وتقنيات التدريب ، ولكنها تشمل أيضًا الاحتفاظ بالمعرفة والتنظيم.

المجالات المتعلقة بالتعلم الآلي مدى الحياة

هناك العديد من المجالات المتعلقة بالتعلم الآلي مدى الحياة ، بما في ذلك **التعلم الانتقالي ، والتعلم متعدد المهام ، والتعلم اللامتناهي ، والتعلم الذاتي ، والتعلم الاونلاين ، وتعلم العالم المفتوح**. بشكل عام ، يمكننا اعتبارها أنواعًا مختلفة من LML ، مع تركيز كل منها على قضايا فرعية محددة. في الأقسام التالية ، نصف بإيجاز كل منهم.

التعلم الانتقالي

تم إجراء بحث مكثف حول التعلم الانتقالي في السنوات الأخيرة. بشكل عام ، يتكون التعلم الانتقالي من مجالين: مجال مصدر ومجال مستهدف. يحتوي المجال المصدر على كمية جيدة من بيانات التدريب المسمى ، بينما يحتوي المجال الهدف على القليل من بيانات التدريب أو لا يحتوي على أي بيانات. الهدف هو استخدام المعلومات الخاضعة للإشراف من المجال المصدر للمساعدة في توقع المجال الهدف. بعبارات أبسط ، يساعد التعلم الانتقالي في عملية تعلم مهمة معينة من خلال الاستفادة من معرفة مجال آخر. التعلم الانتقالي هو حالة خاصة لـ LML. لأنه عادة ما يخزن فقط بيانات المجال المصدر. يفترض التعلم الانتقالي أيضًا عادةً أن المجال المصدر والمجال الهدف مرتبطان ارتباطًا وثيقًا.

التعلم متعدد المهام

التعلم متعدد المهام هو تعلم العديد من المهام ذات الصلة في وقت واحد ، بهدف تحقيق أداء أفضل باستخدام المعلومات ذات الصلة التي تشاركها المهام. يؤدي تعدد المهام أيضًا إلى تجنب الضبط الزائد في مهمة واحدة ، مما يؤدي إلى تعميم أفضل. تركز المهام المتعددة عادةً على تقليل الأخطاء في جميع المهام ، لذلك عند إدخال مهمة جديدة ، يجب تنفيذها في جميع المهام ، بما في ذلك جميع المهام السابقة. من ناحية أخرى ، يستخرج LML المعرفة من المهام السابقة ويجمعها وينطبق فقط على العمل الجديد باستخدام المعرفة المحفوظة. على غرار التعلم الانتقالي ، يفترض تعدد المهام عادةً أن المهام مترابطة.

التعلم اللانهائي[1]

التعلم اللانهائي له نفس منطق LML من حيث أنه يهدف إلى تحقيق أداء أفضل بعد عرض المزيد من البيانات. نظام التعلم اللامتناهي الأكثر شهرة هو متعلم اللغة اللانهائي[2] (NELL) ، والذي يهدف إلى استرداد المعلومات من الويب لبناء قاعدة معرفية منظمة. كل يوم ، هدف التعلم هو تحقيق أداء أفضل من اليوم السابق.

التعلم الذاتي[3]

التعلم الذاتي هو نوع خاص من التعلم الانتقالي يكون فيه المجال المصدر هو نفسه المجال الهدف. لذلك فهو يركز فقط على مجال واحد. يتم الحصول على المعرفة من كميات كبيرة من البيانات غير المسماة (البيانات الوصفية) التي يكون الحصول عليها أسهل بكثير من الحصول على البيانات المصنفة. يتم تمثيل البيانات المصنفة والبيانات غير المسماة بـ D_L و D_U، على التوالي. لا يوجد افتراض حول العلاقة بين D_U و D_L. يمكن أن يكون لـ D_U توزيع مولد مختلف عن D_L. لا يحتاج D_U إلى علامات D_L.

الخطوات الأساسية للتعلم الذاتي هي كما يلي:

1. **تعلم تمثيل عالي المستوى لـD_U.**
2. **تعيين هذه الميزات التجديدية[4] لـD_L.**

---

[1] Never-ending Learning

[2] Never-Ending Language Learner

[3] Self-Taught Learning

[4] Regenerate

3. **بناء نموذج تعليمي خاضع للإشراف (على سبيل المثال ، SVM) على الميزات المعاد إنشاؤها من الخطوة 2.**

منطق التعلم للتمثيل عالي المستوى لـ D_U هو أنه من خلال كميات كبيرة من البيانات غير المسماة ، قد تكون الخوارزمية قادرة على تعلم "العنصر الأساسي" الذي يتكون من كائن. على سبيل المثال ، بالنسبة للصور ، يمكن أن تكون السمة الرئيسية لـ D_L هي قيم كثافة البكسل. من خلال تعلم البيانات غير المسماة ، قد تتعلم الخوارزمية عرض الصور باستخدام الحواف على الصور بدلاً من قيم كثافة البكسل الأولية. من خلال تطبيق هذا التمثيل المكتسب على D_L ، نحصل على تمثيل بمستوى أعلى لـ D_L والذي من المتوقع أن يكون أكثر قابلية للتعميم. بعد تعلم التمثيل غير المراقب ، يتم تحويل كل عينة تعليمية رئيسية إلى مساحة بعد جديدة ، ويمكن إنشاء خوارزمية تعلم خاضعة للإشراف ، مثل SVM ، باستخدام البيانات التعليمية المحولة.

التعلم الاونلاين[1]

تمت دراسة التعلم الاونلاين على نطاق واسع في مجتمع التعلم الآلي. وتتمثل مهمتها في التعلم من التدفق المستمر للبيانات. تكوين التعلم الاونلاين مشابه لـ LML. بمعنى أنه يعمل في سيناريو دفق البيانات. لكن التعلم الاونلاين يفترض عادةً أن البيانات الجديدة تشترك في نفس التوزيع مثل البيانات الحالية ، بينما يفترض LML أيضًا أن البيانات الجديدة قد تنتج من مهمة جديدة ليس لها نفس التوزيع (أو حتى غير ذي صلة).

تعلم العالم المفتوح[2]

يتعامل تعلم العالم المفتوح مع مشكلة تحديد فئات جديدة في وقت الاختبار ، وبالتالي تجنب الإسناد غير الصحيح للفئات المعروفة. عندما يتم دمج فئات جديدة في النموذج ، فإنه يحل مشكلة التعلم مدى الحياة. بهذه الطريقة ، يمكن اعتبار تعلم العالم المفتوح مهمة فرعية للتعلم مدى الحياة.

---

[1] Online learning

[2] Open world learning

التعلم الآلي مدى الحياة للنماذج المميزة

تم إجراء بحث جيد حول LML فيما يتعلق بقضايا التصنيف. يمكن تصنيف العمل السابق إلى ثلاث فئات:**المناهج الهجينة المعيارية**[1]، **والمناهج القائمة على التنظيم**[2]، **ومناهج المراجعة**[3].

المناهج الهجينة المعيارية

تستخدم المناهج الهجينة المعيارية مكونات أو معاملات نموذجية مختلفة لمهام مختلفة لمنع نسيان النموذج. أبسط منهج معياري هو تعليم نموذج منفصل لكل مهمة. ومع ذلك ، فشل هذا المنهج في استخدام النماذج المدربة مسبقًا وبالتالي يؤدي إلى إهدار الموارد. لحل هذه المشكلة، تم اقتراح **شبكات تقدمية**[4] باستخدام المعرفة المسبقة للنماذج المدربة مسبقًا. افترض أن شبكة عصبية تحتوي على L طبقة $\{h_i^{(1)}\}_{i=1}^{L}$ تم تدريبها في المهمة 1. عند وصول مهمة جديدة ، يتم إصلاح معاملات النموذج الذي تم تدريبه مسبقًا ، ويتم تقديم شبكات فرعية جديدة وتطبيقها على خرائط الميزات السابقة لإنشاء خرائط الميزات لهذه المهمة الجديدة.

المناهج القائمة على التنظيم

على الرغم من أن الأساليب الهجينة المعيارية يمكن أن تحل مشكلة النسيان الكارثي ، إلا أنها تتطلب الكثير من الذاكرة ؛ هناك حاجة إلى عدة وحدات أو حتى وحدة واحدة لكل مهمة للمساعدة في تعلم جميع المهام. نوع آخر من النهج هو الضبط الدقيق[5] للنموذج الذي تم تدريبه مسبقًا للمهام الجديدة وتوصيل الشبكة للحفاظ على أداء النموذج في المهام السابقة. عادةً ما يحتفظ هذا النوع من نهج التنظيم بنموذج لجميع المهام أو يضيف معاملات محدودة جدًا لكل مهمة جديدة. هناك طريقتان لتعيين معاملات الشبكة. الطريقة الأولى هي معاقبة معاملات الشبكة المهمة للمهام السابقة من التغيير عند تعلم المهام التالية. الطريقة الثانية هي تخصيص معاملات الشبكة بحيث لا تتغير مخرجات النماذج المدربة مسبقًا عند تعلم المهام التالية.

مناهج المراجعة

تتضمن مناهج المراجعة مخزن ذاكرة مؤقت يخزن عددًا صغيرًا من العينات للمهام السابقة. يُقترح الاحتفاظ بمجموعة فرعية من العينات في كل فئة يمكنها تقدير متوسط كل فئة على أفضل

[1] modular compositional approaches

[2] regularization based approaches

[3] rehearsal approaches

[4] Progressive Networks

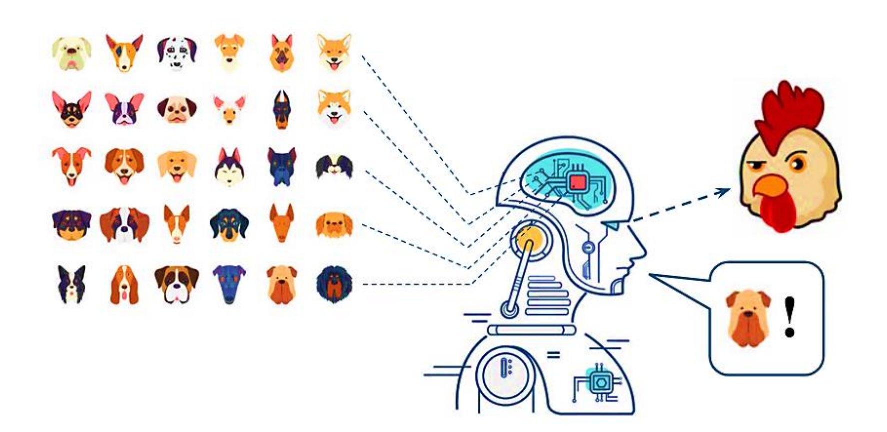
[5] fine-tune

وجه. طريقة المراجعة عرضة للضبط الزائد للعينات المخزنة. لمعالجة هذه المشكلة ، تم اقتراح **مناهج تستند إلى ذاكرة التدرج العرضي**[1] (GEM) مؤخرًا. الغرض من GEM هو تحديث معاملات النموذج المدرب حتى لا تزداد الخسارة لجميع المهام السابقة.

اكتشاف خارج التوزيع[2]

غالبًا ما يتم تدريب الشبكات العصبية العميقة على افتراضات العالم المغلق ، أي أنه يُفترض أن يكون توزيع البيانات التجريبية مشابهًا لتوزيع بيانات التدريب. ومع ذلك ، عند استخدامه في العمل الواقعي ، يكون هذا الافتراض غير صحيح ويؤدي إلى انخفاض كبير في أدائهم. على الرغم من أن هذا الانخفاض في الأداء مقبول لتطبيقات مثل التوصية بالمنتجات ، إلا أن استخدام مثل هذه الأنظمة في مجالات مثل الطب والروبوتات المنزلية يعد أمرًا خطيرًا ، حيث يمكن أن يتسببوا في حوادث خطيرة.

عندما تعالج الشبكات العصبية العميقة البيانات التي لا تشبه التوزيع الذي لوحظ أثناء التدريب (ما يسمى خارج التوزيع) ، فإنها غالبًا ما تقدم تنبؤات خاطئة وتقوم بذلك بثقة كبيرة (الشكل 9ـ3). انظر). ينبغي ، إن أمكن ، تعميم نظام الذكاء الاصطناعي المثالي على عينات خارج التوزيع. لذلك ، فإن القدرة على اكتشاف الخروج من التوزيع أمر بالغ الأهمية للعديد من تطبيقات العالم الحقيقي.



الشكل 9ـ3. عندما يتم تقديم عينة جديدة خارج التوزيع المكتسب ، تتنبأ الشبكات العصبية بفئة من التوزيع المكتسب بثقة كبيرة. يقترح اكتشاف خارج التوزيع خوارزميات لحل هذه المشكلة.

[1] Gradient Episodic Memory

[2] Out-of-distribution detection

يعد اكتشاف التوزيع غير الضروري لضمان موثوقية وسلامة أنظمة التعلم الآلي. في القيادة الذاتية ، على سبيل المثال ، نريد أن يقوم نظام القيادة بالتحذير وتسليم السيطرة عندما يكتشف مشاهد أو أشياء غير عادية لم يراها من قبل ولا يمكنه اتخاذ قرار آمن.

ظهرت هذه المشكلة لأول مرة في عام 2017 ومنذ ذلك الحين جذبت اهتمامًا متزايدًا من مجتمع البحث. يعتمد معظم العمل الأخير على الكشف عن خارج التوزيع على تدريب مراقبة الشبكات العصبية التي تعمل على تحسين **خطأ الانتروبيا المتبادلة**[1]. في هذه الحالات ، يرتبط إخراج الشبكة ارتباطًا مباشرًا بحل المشكلة ، أي احتمال كل فئة. ومع ذلك ، يجب أن يكون مجموع تمثيلات متجه الإخراج واحدًا دائمًا. هذا يعني أنه عندما يتم عرض إدخال على الشبكة ليس جزءًا من توزيع التدريب ، فإنه لا يزال يعطي الاحتمال لأقرب فئة بحيث يصل مجموع الاحتمالات إلى واحد. أدت هذه الظاهرة إلى المشكلة المعروفة المتمثلة في **شبكات عصبية آمنة للغاية** [2] لمحتوى لم يسبق رؤيته من قبل.

التعليم المعزز[3]

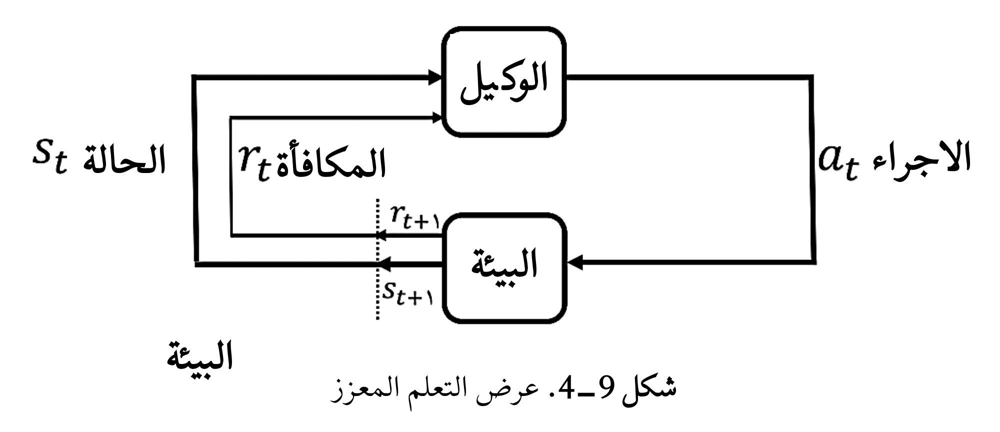
على الرغم من أن التعلم العميق لديه القدرة على تمثيل البيانات بقوة وأداء أفضل من العديد من الأساليب الأخرى في العديد من مسائل التصنيف ومعالجة الصور ، إلا أنه لا يكفي لبناء نظام ذكي للذكاء الاصطناعي. هذا لأن نظام الذكاء الاصطناعي يجب ألا يكون قادرًا على التعلم من البيانات فحسب ، بل يجب أن يتعلم أيضًا من التفاعلات البشرية مع بيئة العالم الحقيقي. يعد التعليم المعزز أحد مجالات التعلم الآلي ويركز على تمكين الجهاز من التفاعل مع بيئة العالم الحقيقي.

يحاول التعليم المعزز من خلال الوكيل حل المشكلة عن طريق التجربة والخطأ من خلال التفاعل مع بيئة غير معروفة للوكيل. يمكن للوكيل تغيير حالة البيئة من خلال أفعاله أثناء إجراء ردود فعل فورية من البيئة. غالبًا ما يشار إلى الاستجابات على أنها **المكافأة** في التعليم المعزز. يكتسب الوكيل القدرة على التعلم بشكل أفضل من خلال تلقي المزيد من المكافآت الإيجابية من البيئة. بشكل عام ، هدف الوكيل هو إيجاد سلسلة الإجراءات المثلى لحل المشكلة. عادة ما يتم نمذجة التعليم المعزز كعملية ماركوف لاتخاذ القرار ويمكن وصفه بالشكل 9ـ4.

[1] cross-entropy loss

[2] overconfident

[3] reinforcement learning



شكل 9ــ4. عرض التعلم المعزز

كما يتضح من الشكل 9ــ4، فإن وحدة التعلم، التي تسمى الوكيل، تغير بيئتها بنشاط عن طريق تحديد الإجراء a من مجموعة من الإجراءات الممكنة. بعد العملية، تتغير البيئة وفقًا لذلك وتخبر الوكيل بالحالة الجديدة. بالإضافة إلى ذلك، ترسل البيئة إشارة مكافأة r إلى الوكيل، مما يوفر ملاحظات حول الإجراء المحدد وفقًا للسيناريوهات المحددة. يستمر الوكيلفي العمل، والانتقال من حالة إلى أخرى، حتى يصل إلى حالة نهائية.

يمكن سرد مكونات نظام التعليم المعزز على النحو التالي:

- **الوكيل:** برنامج تم تدريبه للقيام بمهمة محددة.
- **البيئة:** العالم الحقيقي أو الافتراضي، حيث يتخذ الوكيل الإجراءات.
- **الإجراء:** حركة يقوم بها الوكيل تؤدي إلى تغييرفي الحالة (الحالة)في البيئة.
- **المكافأة:** تحدد دالة المكافأة هدفًافي مشكلة التعليم المعزز وترسم كل حالة ملحوظة من البيئة إلى رقم واحد يشير إلى الرغبة الكامنةفي تلك الحالة. الغرض من المكافأةفي التعليم المعزز هو تقييم إجراء يمكن أن يكون إيجابيًا أو سلبيًا والاستجابات التي يتلقاها الوكيل من البيئة بعد كل إجراء. قد لا تكون المكافأة الحقيقية لفعل الشيء الصحيحفي موقف معين فورية.
- **الحالة:** جميع المعلومات التي يمتلكها الوكيلفي بيئته الحالية.في لعبة الشطرنج، على سبيل المثال، يكون الموضع هو موضع كل القطع على رقعة الشطرنج.
- **المشاهدات:** بالنظر إلى أنهفي بعض الحالات، لا يتمكن الوكيل من الوصول إلى الحالة الكاملة للبيئة، فعادة ما تكون المشاهدة جزءًا من الحالة التي يمكن للعامل مراقبتها. بمعنى آخر، المشاهدات هي المعلومات التي توفرها البيئة للوكيل وتوضح ما يحدث حول الوكيل. ومع ذلك، غالبًا ما يتم استخدامها بالتبادلفي الأدب والوضع والمشاهدة.
- **السياسة:** تحدد ما سيفعله الوكيل وفقًا للوضع الحالي.في مجال التعلم العميق، يمكننا تدريب شبكة عصبية لاتخاذ هذه القرارات. خلال فترة التدريب، يحاول الوكيل تعديل سياسته لاتخاذ قرارات أفضل. تسمى مهمة إيجاد السياسة المثلى تحسين السياسة (التحكم) وهي واحدة من القضايا الرئيسيةفي التعلم المعزز.

- **دالة القيمة:** تحدد ما هو جيد للوكيل على المدى الطويل. بمعنى آخر، عندما نطبق دالة القيمة على حالة معينة، إذا بدأنا من تلك الحالة، فإنها تعطينا العائد الإجمالي الذي يمكن توقعهفي المستقبل.

ستساعدك الأمثلة البسيطة التالية على فهم آلية التعليم المعزز بشكل أفضل:

افترض أن لديك قطة وتريد تدريبها على القيام بأشياء معينة. نظرًا لأن القطة لا تفهم اللغة العربية أو أي لغة أخرى ، فلا يمكن إخبارها مباشرة بما يجب أن تفعله. بدلا من ذلك ، يمكن اتباع استراتيجية مختلفة. نقدم موقفًا وتحاول القطة الرد بطرق مختلفة. إذا كانت استجابة القطة هي الاستجابة الصحيحة ، فإننا نعطيها سمكة. الآن ، عندما تتعرض قطة لنفس الموقف ، تنتظر القطة بفارغ الصبر نفس المكافأة (الطعام). لأنه تعلم أنه إذا قام بعمل معين ، فسوف يكافأ.

في هذا المثال:

- القطة هي التي تكشف البيئة المنزلية في هذه الحالة.
- يمكن أن يكون الموقف عبارة عن جلوس قطة ، ويمكنك استخدام قول كلمات معينة لجعل القطة تمشي.
- من خلال القيام بعمل ما ، يتفاعل الوكيل بالانتقال من حالة إلى أخرى. على سبيل المثال، تنتقل القطة من الجلوس إلى المشي.

يمكن تقديم مثال آخر للأطفال. غالبًا ما يرتكب الأطفال أخطاء. يحاول الكبار التأكد من أن الطفل قد تعلم من هذا الخطأ ومحاولة عدم تكراره مرة أخرى. في هذه الحالة ، يمكننا استخدام مفهوم التغذية الراجعة. إذا كان الآباء صارمين ، فإنهم يلومون أطفالهم على أي أخطاء ، وهي ردود فعل سلبية. من الآن فصاعداً ، سوف يخطئ الطفل في تذكر أنه فعل شيئًا خاطئًا ، لأنه سيلقي باللوم عليه من قبل الوالدين. ثم هناك ردود فعل إيجابية ، حيث قد يمدحهم الآباء لفعلهم الشيء الصحيح. هنا ، نقوم أو نحاول القيام بعمل صحيح بطريقة معينة.

باختصار ، التعليم المعزز هو نوع من منهجية التعلم التي نقدم فيها ملاحظات للخوارزمية مع مكافآت للتعلم منها ، من أجل تحسين النتائج في المستقبل.

عملية ماركوف لاتخاذ القرار[1]

عملية ماركوف لاتخاذ القرار هي نموذج رياضي عشوائي لسيناريو صنع القرار. في كل مرحلة ، يختار صانع القرار ، أو بعبارة أخرى نفس الوكيل ، إجراءً. في هذا النموذج ، يكون جزء من النتيجة عشوائيًا والجزء الآخر هو نتيجة فعل. تُستخدم عمليات صنع القرار في ماركوف لنمذجة

[1] Markov decision process (MDP)

مجموعة متنوعة من مشكلات التحسين ويمكن حلها من خلال البرمجة الديناميكية والتعليم المعزز.

تشبه عملية صنع القرار في ماركوف مخططًا انسيابيًا به دوائر تمثل المواقف او الحالات. ستظهر أسهم من كل دائرة ، تشير إلى جميع الإجراءات الممكنة التي يمكن اتخاذها من هذا الموقف. على سبيل المثال ، عملية صنع القرار في ماركوف في تمثيل لعبة الشطرنج لها مواقف تشير إلى موقع القطع على رقعة الشطرنج والإجراءات التي تشير إلى الحركات المحتملة بناءً على القطع الموجودة على رقعة الشطرنج

الميزة الرئيسية لعملية صنع القرار في ماركوف هي أن كل حالة يجب أن يحتوي على جميع المعلومات التي يحتاجها الوكيل لاتخاذ قرار مستنير ، وهو مطلب يسمى "**أصول ماركوف**". في الأساس ، تقول أصول ماركوف أنه لا يمكن للمرء أن يتوقع أن يكون للوكيل ذاكرة تاريخية خارج حالته. على سبيل المثال ، توضح الحالة الحالية لرقعة الشطرنج كل الأشياء الأفضل للخطوة التالية ، وليست هناك حاجة إلى الحركات التي تم إجراؤها من قبل لحفظها.

في الممارسة العملية ، لا يتعين على التعلم المعزز محاكاة مشكلة العالم الحقيقي من أجل حل مشكلة ما. على سبيل المثال ، قد تلعب ذاكرتي حول كيفية لعب خصم معين الشطرنج دورًا في عملية اتخاذ القرار الخاصة بي في العالم الحقيقي ، ولكن من الممكن الفوز بلعبة شطرنج من خلال التعليم المعزز دون الحاجة إلى هذه المعلومات.

يتم تحديد عمليات صنع القرار في ماركوف من خلال مجموعة من 5 عناصر < S، A، P، R، γ > ، حيث:

- **S:** مجموعة من الحالات التي تتضمن جميع التمثيلات الممكنة للبيئة.
- **A:** في كل حالة ، توفر البيئة للوكيل مجموعة من الإجراءات في مساحة التشغيل للوكيل للاختيار من تلك الإجراءات. العامل يؤثر على البيئة من خلال الإجراءات.
- **P:** $P = (s, a, \acute{s}) = P_r(s_{t+1} = \acute{s} | s_t = s, a_t = a)$ مصفوفة الانتقال هي احتمال أن تؤدي الاجراء a في الحالة s في الوقت t إلى الحالة $\acute{s}$ في الوقت $t + 1$.
- **R:** $R = (s, a, \acute{s})$ المكافأة المتوقعة التي يتلقاها الوكيل بعد الإجراء a في الحالة s والوصول إلى الحالة $\acute{s}$.
- $\boldsymbol{\gamma}$**:** عامل هبوط ويظهر الأهمية بين المكافآت قصيرة الأجل وطويلة الأجل.

تتمثل إحدى القضايا الرئيسية في عمليات صنع القرار في ماركوف في إيجاد "سياسة" لصانع القرار ؛ دالة P تحدد الحالات للعمليات $a = \pi(s)$. يمكن أن تكون هذه السياسة محددة أو عشوائية. الهدف هو العثور على سياسة تزيد من إجمالي المكافأة إلى الحد الأقصى من حالة إلى أخرى:

$$G_t = \sum_{i=t}^{\infty} \gamma^i . R(s_i, a_i, s_{t+1})$$

يسمى G_t الدخل او العائد.

الوكيل

الوكيل هو شخص أو شيء يتفاعل مع هذه البيئة من خلال تنفيذ إجراءات وملاحظات معينة وتلقي المكافآت النهائية. العامل هو مكون يقرر بناءً على الثواب والعقاب، الإجراء الذي يجب اتخاذه. لاتخاذ قرار، يُسمح للوكيل باستخدام أي مراقبة للبيئة وأي قانون داخلي. يمكن أن تكون هذه القواعد الداخلية أي شيء، ولكن عادةًفي التعليم المعزز يتوقع المرء أن يتسم الوضع الحالي من بيئة الامداد إلى الحالة باتخاذ قرارات ماركوف، ثم يقرر، باستخدام دالة السياسة، ما هو القرار الذي يجب اتخاذه.

في معظم سيناريوهات التعلم المعزز العملية، من المفترض أن يحل وكيل البرمجيات لدينا بعض المشكلات بطريقة أكثر أو أقل كفاءة. الوكيل هو أحد أهم مكونات النظام القائم على التعليم المعزز. لأنه يتضمن ذكاءً لاتخاذ القرار والتوصية بالإجراءات (العمليات) المثلىفي أي موقف. نظرًا لأن الوكيل يلعب دورًا مهمًا للغايةفي التعليم المعزز، فقد تم إجراء الكثير من الأبحاث حول بنية التعلم والنماذج ذات الصلة. فيما يلي، نقسم الخوارزميات بناءً على الوكلاء إلى خوارزميات **قائمة على القيمة، وقائمة على الخط، وقائمة على النموذج.**

خوارزميات قائمة على القيمة

في الخوارزميات القائمة على القيمة، يتم تعيين دوال القيمة $V^{\pi}(s)$ للحالات ويتخذ صانع القرار قراراته بناءً على قيم الحالات. دالة القيمة هي دالة تقيم مدى استناد الموقف إلى التنبؤ بالمكافآت المستقبلية. هناك نوعان مختلفان من دوال القيمة:

- **دالة الحالة_القيمة،** التي يشار إليها عادةً باسم دالة القيمة،G_t هي العائد المتوقع بدءًا من الحالة s وتتبع السياسة π ، ويتم تعريفها بواسطة معادلة بلمان على النحو التالي:

$$V^{\pi}(s) = \mathbb{E}[G_t|S_t = s]$$

$$= \sum_{a\in A} \pi(a|s) \sum_{\acute{s}\in S} \pi(\acute{s}|s,a)[\mathrm{R} = (s,a,\acute{s}) + \gamma^{V^{\pi}(\acute{s})}]$$

- **دالة الحالة_الاجراء ،** يُشار اليها عادةً بقيمة Q ، العائد G_t المتوقع هو زوج من الحالة_ الاجراء في الوقت t للسياسة π ، ويتم تعريفه بالمثل بواسطة معادلة بلمان على النحو التالي:

$$Q^{\pi}(s,a) = \mathbb{E}[G_t|S_t = s, A_t = a]$$
$$= \sum_{\acute{s}\in S} P(\acute{s}|s,a)[\mathrm{R} = (s,a,\acute{s}) + \gamma \sum_{a\in A} \pi(\acute{a},\acute{s}) + Q^{\pi}(\acute{s},\acute{a})]$$

معادلات بلمان

معادلات بلمان هي مجموعة من المعادلات التي تحلل دالة القيمة إلى مكافآت فورية بالإضافة إلى القيم المستقبلية المخصومة.

$V^{\pi}(s) = \mathbb{E}[G_t|S_t = s]$
$= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots |S_t = s]$
$= \mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \ldots)|S_t = s]$
$= \mathbb{E}[R_{t+1} + \gamma G_{t+1}|S_t = s]$
$= \mathbb{E}[R_{t+1} + \gamma V(S_{t+1})|S_t = s]$

وبالمثل بالنسبة لـ Q_القيمة:

$Q^{\pi}(s,a) = \mathbb{E}[R_{t+1} + \gamma V(S_{t+1})|S_t = s, A_t = a]$
$= \mathbb{E}[R_{t+1} + \gamma \mathbb{E}_{a\sim\pi} Q(S_{t+1}, a)|S_t = s, A_t = a]$

تم إنشاء المعادلة التالية بين دالات الحالة_القيمة و الحالة_الإجراء:

$$V^{\pi}(s) = \sum_{a} \pi(a|s)\, Q^{\pi}(s,a)$$

الهدف هو زيادة إجمالي المكافأة التراكمية على المدى الطويل. بمعنى آخر ، الهدف من التعليم المعزز هو إيجاد السياسة المثلى. يُطلق على السياسة الذي تزيد الحد الأقصى للمكافأة التراكمية **السياسة المثلى** ويُشار إليه بعلامة π^*. السياسة المثلى π^* هو أن قيمة كل حالة s تحت π^* أكبر من أو تساوي قيمة الحالة π^* تحت سياسة اخرى π' لجميع $s \in S$:

$$V^{\pi^*}(s) = V^*(s) \geq V^{\pi'}\ \forall\, s \in S, \pi'$$

إذا كانت دالة القيمة_الحالة هي الأمثل، يستخدم الوكيل السياسة المثلى، وقد يكون هناك العديد من السياسات المثلى التي تؤدي إلى نفس دالة القيمة_الحالة المثلى. يمكن تعريف دالة الحالة_القيمة المثلى V^* على النحو التالي:

$$V^*(s) = max_{\pi} V^{\pi}(s)\ \forall\, s \in S$$

بالإضافة إلى ذلك ، تؤدي السياسة المثلى إلى دالةالإجراء_الحالة المثلى Q^*:

$$Q^*(s,a) = max_{\pi} Q^{\pi}(s,a)\ \forall\, s \in S, a \in A$$
$$= \mathbb{E}[R_{t+1} + \gamma V^*(\acute{s})|S_t = s, A_t = a]$$

أخيرًا ، يمكن اشتقاق معادلة تحسين بلمان من المعادلات التي تم تقديمها مسبقًا:

$$V^*(s) = max_{a\in A(s)} Q^{\pi^*}(s,a)$$
$$= max_a\ \mathbb{E}[R_{t+1} + \gamma V^*(\acute{s})|S_t = s, A_t = a]$$
$$= max_a \sum_{\acute{s}\in S} P(\acute{s}|s,a)[\mathrm{R}(s,a,\acute{s}) + \gamma V^*(\acute{s})]$$

$$= max_a \sum_{\acute{s}\in S} P(\acute{s}|s,a)[\text{R}(s,a,\acute{s}) + \gamma max_{\acute{a}} Q^{\pi^*}(\acute{s},\acute{a})]$$

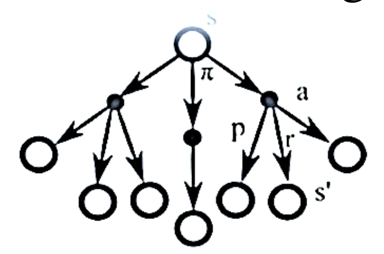
مخطط الدعم في التعلم المعزز

يمكن فهم معادلات بلمان بشكل أفضل باستخدام شجرة الحالات والاجراءات (مخطط الدعم). يُظهر مخطط الدعم تمثيلًا مرئيًا للخوارزميات والنماذج المختلفةفي التعليم المعزز.

عملية الدعم (عملية التحديث) هي تمثيل رسومي للخوارزمية، تشير إلى الحالة، والإجراء، وحالة التحويل، والمكافأة، وما إلى ذلك. يوضح هذا الرسم البياني قيمة الحالة بدائرة مجوفة بينما يتم عرض قيمة الإجراء مع دائرة صلبة. أيضًا، يتم عرض الإجراء بسهم يبدأ من الحالة.

الآن دعونا ننظرفي كيفية عرض قيمة الحالة باستخدام مخططات الدعم (الشكل 9ـ5).

1. s هو وضع البداية والعقدة الرئيسية.
2. من الوضع s، يمكن إجراء ثلاث عمليات. كما يتضح من السهم ويتصرف الوكيل وفقًا لسياسة π.
3. إذا كانت البيئة العشوائية لديها احتمالية معينة للإرسال، فيمكن أن ينتهي هذا العاملفي مواقف مختلفة. كما هو موضحفي الشكل، هناك 3 أوضاع محتملة يمكن للوكيل إدخالها بعد تنفيذ الإجراء الصحيح.



الشكل 9ـ5. مخطط الدعم للحالةـ القيمة $V^{\pi}(s)$

الخوارزميات المستندة إلى السياسة

السياسة هي استراتيجية تمكّن آلية التعلم من تحديد الإجراء التالي الأفضل بناءً على الوضع الحالي. السياسة هي جوهر التعلم المعزز؛ أي أنه وحده يكفي لتحديد السلوك.

تتخذ الخوارزميات المستندة إلى السياسة نهجًا مباشرًا أكثر من الخوارزميات المستندة إلى القيمة.في هذا النهج، بدلاً من العثور على قيمة كل موقف محتمل ثم الحصول على السياسة المثلى، يسعون للعثور على السياسة مباشرةً لتعظيم العائد المتوقع.

وبعبارة أخرى، فإن جوهر الأساليب المستندة إلى السياسة هو تحديث معاملات السياسة θ ؛ بحيث يزيد عائد G_t المتوقع. بعبارات أبسط،في النهج القائم على السياسة،

يتم أولاً اختيار سياسة عشوائية وفي مرحلة التقييم يتم العثور على دالة القيمة لتلك السياسة. بعد ذلك، حساب السياسة الجديدة باستخدام دالة القيمة للتحسين في كل خطوة. يكرر هذه العملية حتى يتم العثور على النهج الأمثل. يتم وصف عملية التحسين على النحو التالي:

$$\theta_{i+1} = \theta_i + \Delta\theta_i$$

حيث θ_i هي مجموعة من معاملات السياسة التي تم تحديد معاملاتها في سياسة π،و $\Delta\theta_i$ هي تغيير معاملات السياسة.

الاستخراج[1] مقابل الاستكشاف [2]

يحاول البشر الحصول على أكبر قدر ممكن من المعلومات قبل القيام بأي شيء. على سبيل المثال، قبل تجربة مطعم جديد، حاول قراءة التعليقات أو اسأل الأصدقاء الذين جربوه بالفعل. من ناحية أخرى،في التعليم المعزز، هذا غير ممكن. ولكن هناك بعض الأساليب التي يمكن أن تساعدك في اكتشاف أفضل استراتيجية.

في التعليم المعزز، يمكن للوكيل تطبيق استراتيجيتين عند اتخاذ القرارات في البيئة:

- **الاستكشاف:** اختر إجراءً عشوائيًا. باتباع هذه الطريقة، يمكن للوكيل زيارة الحالات الجديدة والعثور على سياسات جديدة وأفضل. والنتيجة هي المزيد من المعلومات التي قد تؤدي إلى قرارات أفضل في المستقبل.
- **الاستخراج:** كن طماعًا. اختيار أفضل الأعمال المعروفة حسب العلم المتاح، باستخدام المكافأة الإجمالية. بمعنى آخر، تعتزم اتخاذ القرار الأفضل بناءً على المعلومات الحالية.

في التعليم المعزز، يتم استخراج هذا النوع من اتخاذ القرار: عندما تستمر في العمل السابق، يسمى الاستخراج، وعندما تحاول أشياء أخرى، يُطلق عليه الاستكشاف.

إحدى السمات الرئيسية للتعليم المعزز هي مشكلة الاستخراج مقابل الاستكشاف. إذا أراد الوكيل تعلم أفعال أفضل، أو بعبارة أخرى، أفعال ستؤدي في النهاية إلى المزيد من المكافآت المتراكمة، فعليه أن يجرب أفعالًا جديدة. أيضًا، إذا كان الوكيل قد استخرج من معرفته الحالية واتبع الإجراءات المعروفة سابقًا للحصول على تعليقات جيدة للمكافأة، فلا يمكن ضمان الحصول على عائد أعلى من المكافأة التي كان من الممكن أن يحصل عليها الوكيل. إذن فهذه معضلة يواجهها الوكيل عند اتخاذ قرار بشأن المزيد من الإجراءات: إما تجربة إجراءات عشوائية وتخمين أنه سيحصل على المزيد من المكافآت، ولكن يخاطر بتفاقم النتيجة. أو من خلال التصرف وفقًا لظروفه الحالية، فمن المحتمل أن يحصل على مكافأة أقل ولكن مؤكدة. بمعنى

---

[1] exploitation

[2] exploration

آخر، إذا كان الوكيل يقوم بالاستكشاف فقط، فقد لا يحقق درجات أعلى ويحسن أفعاله. من ناحية أخرى، إذا تم استخدام الاستخراج فقط، فقد يعلقفي سياسته الحالية من خلال رؤية جميع المسارات الممكنة. وبالتالي، من المرجح أن يفوت الوكيل السياسة المثلى. لذلك، يجب أن يكون هناك توازن جيد بين الاستكشاف والاستخراج.

تنشأ هذه المشكلة لأن عملية التعلمفي التعليم المعزز تتم اونلاين. بمعنى آخر، لا تتم مراقبة التعليم المعزز بقدر ما يتم مراقبة التعليم. لذلك، يسعى الوكيل نفسه إلى جمع البيانات بطريقة ما ويؤثر على البيانات المرصودة من خلال الإجراءات التي يقوم بها. لذلك، من المفيد أحيانًا القيام بإجراءات مختلفة للحصول على بيانات جديدة.

مشكلة الاستكشاف مقابل الاستخراج هي موضوع متكررفي التعليم المعزز والذكاء الاصطناعي بشكل عام. هل علينا أن نستخلص من المعرفة المكتسبة، أي هل علينا أن نتبع مسارًا معروفًا بمكافآت كبيرة؟ أم ينبغي علينا استكشاف المواقف المجهولة بحثًا عن سياسة جديدة أفضل؟ التوازن بين كل من استراتيجيات صنع القرار يحسن بشكل كبير أداء التعلم للوكيل. الإجابة المقبولة لحلها هي أن أحد العوامل يحتاج أولاً إلى اكتشاف معظم المواقف ثم الحصول على نتائج أفضل من خلال استخراج المعرفة المتراكمة. لكن يجب ألا ننسى أنهفي البيئات الديناميكية، من الصعب للغاية الحصول على ما إذا كان قد تم الاستكشاف الكافي أم لا. ومع ذلك، هناك عدة طرق لاختيار الإجراءات الموضحة أدناه.

الاستخراج: يعني تحديد أداء الوكيل باستخدام المعرفة الموجودة، والتي يتم عادةً تقييم أدائها بالمكافأة المتوقعة. على سبيل المثال، أصبح لمنقب الذهب الآن خامًا يزوده بغرامين من الذهب يوميًا، وهو يعلم أن أكبر خام الذهب يمكن أن يعطيه خمس غرامات من الذهب يوميًا. ومع ذلك، فهو يعلم أيضًا أن العثور على خام جديد لن يجبره على التوقف عن تعدين الخام الحالي فحسب، بل سيتكبده أيضًا تكاليف إضافية لأنه يخاطرفي النهاية بعدم العثور على أي شيء. مع وضع ذلكفي الاعتبار، قرر التنقيبفي الخام الحالي لتعظيم الحد الأقصى من المكافأة (في هذه الحالة الذهب) من خلال الاستخراج، ويتخلى عن الاستكشاف بسبب المخاطر الكبيرة للاستكشاف. السياسة التي تبناها هنا هي سياسة جشعة، مما يعني أن الوكيل يؤدي باستمرار الإجراء الذي ينتج عنه أعلى مكافأة متوقعة بناءً على المعلومات الحالية، بدلاً من إجراء تجارب محفوفة بالمخاطر تقلل من المكافآت المتوقعة.

الاستكشاف: يعني زيادة المعرفة الحالية بالأفعال والتفاعل مع البيئة. بالعودة إلى منقب الذهب، على سبيل المثال، فإنه يتمنى أن يقضي بعض الوقتفي البحث عن خام جديد، وإذا وجد خامًا ذهبيًا أكبر، فيمكنه الحصول على المزيد من المكافآت كل يوم. للحصول على عوائد طويلة الأجل، قد يتم التضحية بالعوائد قصيرة الأجل. يواجه منجم الذهب دائمًا مشكلة الاستخراج والاستكشاف، لأنه يتعين عليه تحديد مقدار منجم الذهب الذي يجب أن يبقى

فيه ومقدار منجم الذهب الذي يجب أن يستمرفي العمل. يصف التفاعل بين الاستكشاف والاستخراج التوازن بين مقدار جهد العاملفي الاستكشاف والاستخراج، على التوالي. يعتبر التفاعل بين الاستكشاف والاستخراج موضوعًا رئيسيًا لبحوث التعليم المعزز وتطوير خوارزميات التعلم المعزز.

إستراتيجية اختيار الجشع - إبسيلون

طريقة ممكنة وبسيطة وشائعة لكنها فعالة لاختيار الإجراءفي كل مرحلة كاستراتيجية اختيار هي اختيار الجشع ـ إبسيلون. في هذه الطريقة، تتحكم المعامل e بين 0 و 1في العامل الذي يعمل أثناء قيامه بالاستخراج أو الاستكشاف. باستخدام هذه الطريقة،في أي وقت، من المحتمل أن يختار الوكيل بين الاستكشاف والاستخراج. يستكشف مع الاحتمال e والاختيار العشوائي من جميع العمليات المتاحة ومع احتمال $1-e$.

تؤدي القيم العالية لـ e إلى قيام الوكيل بمزيد من البحث، ونتيجة لذلك، تقلل من احتمالية التطبيق الأمثل. حيث أنه يمنح الوكيل القدرة على الاستجابة بسرعة للتغييرات التي تحدثفي البيئة.في المقابل، تؤدي القيم المنخفضة للعامل e إلى إجراءات أكثر كفاءة.

استكشاف بولتزمان

طريقة أخرى لتحديد الإجراءات هي سياسة توزيع بولتزمان. توزيع بولتزمان هو سياسة تعليمية تقلل من الميل للاستكشاف بمرور الوقت. ومن المفترض أنه مع تقدم التعلم، يتحسن النموذج الحالي. يخصص توزيع بولتزمان احتمالية لكل عملية باستخدام معامل T تسمى درجة الحرارة.

توزيع بولتزمان، باستخدام المعادلة التالية، يخصص احتمالية إيجابية لكل إجراء محتمل $a\epsilon A$:

$$P(a|s)=\frac{e^{\frac{Q(s,a)}{T}}}{\sum \acute{a}\epsilon A e^{\frac{Q(s,\acute{a})}{T}}}$$

حيث:

$$T_{\text{الجديدة}}=e^{-dj}*T_{\text{الحد الاكثر}}+1$$

يرتبط الاجراء مع المزيد من $Q(s,a)$ باحتمال أكبر لـ P. يتناقص T مع زيادة التكرار j بمرور الوقت. وهكذا ، مع تقدم التعلم ، يتناقص الميل للاستكشاف في الوكيل ، ونتيجة لذلك ، تميل سياسة التعلم لتوزيع بولتزمان إلى أن تكون مشتقة من الاجراءات ذات $Q(s,a)$ العالية. تُضبط المعاملات $T_{\text{الحد الاكثر}}$ومعدل الهبوط dj أولاً.

بناءً على النموذج مقابل بدون نموذج

كما ذكرنا سابقًا ، فإن الهدف الرئيسي للوكيل في التعلم المعزز هو تجميع أكبر قدر من المكافأة على "المدى الطويل". للقيام بذلك ، يجب على الوكيل إيجاد السياسة المثلى للسلوك في البيئة. يمكن أن تكون البيئة محددة أو عشوائية (غير مؤكدة). بمعنى ، إذا قام الوكيل بإجراء ما في موقف معين ، فقد لا تكون الحالة التالية للبيئة هي نفسها دائمًا. من المؤكد أن هذه الشكوك تجعل من الصعب العثور على السياسة المثلى.

التنبؤ مقابل المراجعة

تشير إمكانية التنبؤ والمراجعة إلى مسألتين أساسيتين يجب على عامل التعلم المعزز معالجتهما. يتطلب التنبؤ القدرة على حساب أو تقدير عواقب الإجراء. عادة ما تعني المشاكل التنبؤية تقدير قيم الحالات أو قيم الاجراء لأزواج الحالة_الاجراء.في المقابل، تتطلب المراجعة القدرة على اتخاذ القرارات. بدون مراجعة، الوكيل لا يتخذ أي إجراء.

مشكلة التنبؤ: بالنظر إلى عملية صنع القرار لماركوف $< S, A, P, R, \gamma >$ والسياسة π ، يجب إيجاد دالة القيمة $v\ (\pi)$. بعبارة أخرى، الهدف هو فهم مدى جودة السياسة.

مراجعة المشكلة: وفقًا لعملية صنع القرار الخاصة بماركوف $< S, A, P, R, \gamma >$ ، يجب العثور على القيمة المثلى لـ $v\ (\pi)$ والسياسة المثلى π^*. بمعنى آخر، الهدف هو العثور على السياسة التي توفر أكبر قدر من المكافأة مع أفضل إجراء للاختيار من بينها.

كما نعلم ، في التعلم المعزز ، غالبًا ما تتم المشكلة من وجهة نظر رياضية كعملية صنع قرار ماركوف. عملية صنع القرار في ماركوف هي طريقة لإظهار " ديناميكيات" البيئة ؛ أي كيف تتفاعل البيئة مع الإجراءات المحتملة للوكيل في موقف معين. بتعبير أدق ، تم تجهيز عملية صنع القرار في ماركوف بدالة نقل ، وهي دالة ، نظرًا للحالة الحالية للبيئة والعلم الذي قد يؤديه الوكيل ، تجعل من الممكن التحويل إلى أي من المخرجات. ترتبط دالة المكافأة أيضًا بعملية صنع القرار في ماركوف.

تتم مكافأة دالة المكافأة وفقًا للحالة الحالية للبيئة وربما الإجراء الذي يقوم به الوكيل والحالة التالية للبيئة. غالبًا ما تسمى دوال المكافأة والتحويل بأنماط البيئة. ومع ذلك ، في بعض الأحيان ليس لدينا دوال المكافأة والتحويل. ومن ثم لا يمكننا تقدير السياسة لأنها غير معروفة. في غياب هذه الدوال ، لتقدير السياسة المثلى ، من الضروري التفاعل مع البيئة ومراقبة استجاباتها ، والتي يشار إليها غالبًا باسم "مشكلة التعليم المعزز". لأنه ، يجب على الوكيل تقدير السياسة من خلال تعزيز معتقداته حول ديناميكيات البيئة.

بمرور الوقت ، يبدأ الوكيل في فهم كيفية تفاعل البيئة مع أفعاله ويمكنه تقدير السياسة المثلى. لذلك ، في مشاكل التعليم المعزز ، يتم تقدير عامل السياسة الأمثل للسلوك في بيئة غير مألوفة

من خلال التفاعل معها باستخدام طريقة "التجربة والخطأ". وفقًا لذلك ، يمكن تقسيم خوارزميات التعلم المعزز إلى خوارزميات قائمة على النموذج أو غير قائمة على النموذج.

في الخوارزميات المستندة إلى النموذج، يمكن للوكيل الوصول إلى نموذج كامل للبيئة، أو يحاول تعلمه من خلال التفاعل، ويستخدم دالة النقل والمكافأة لتقدير السياسة المثلى بدقة. بمعنى آخر، يحاول الوكيل أخذ عينات من النموذج الاحتمالي وتعلمه واستخدامه لتحديد أفضل الاجراءات. يجوز للوكيل فقط الوصول إلى دوال التحويل والمكافأة التقريبية التي تعلمها الوكيل. حيث أنه يتفاعل مع البيئة أو يمكن أن يعطى لوكيل واحد من خلال وكيل آخر، على سبيل المثال.

بشكل عام،في الخوارزمية القائمة على النموذج، يمكن للوكيل توقع ديناميكيات البيئة أثناء أو بعد مرحلة التعلم. هذا لأنه يحتوي على تقدير لدوال النقل ودالة المكافأة، وإذا تم التعرف على احتمالية النقل بنجاح، فإن الوكيل يعرف مدى احتمالية الدخولفي حالة معينة بالنظر إلى الحالة الحالية والإجراء. وتجدر الإشارة، مع ذلك، إلى أن دوال التحويل والمكافأة التي يستخدمها الوكيل لتحسين تقدير سياسته المثلى قد تكون مجرد تقريب للدوال "الحقيقية". ومن ثم قد لا يتم العثور على السياسة المثلى بسبب هذه التقريبات.

على عكس الخوارزميات القائمة على النموذج، لا تمتلك الخوارزميات غير النموذجية معرفة أساسية بدالة النقل ويجب أن تتعلمها أثناء التعلم لإيجاد مسارات فعالة. بعبارة أخرى، تقدر الخوارزمية غير المعدلة إما "دالة القيمة" أو "دالة السياسة" مباشرة من التجربة، أي من خلال التفاعل بين الوكيل والبيئة، دون استخدام دوال النقل والمكافأة.

تتمثل إحدى طرق التمييز بين الأساليب القائمة على النموذج وغير القائمة على النموذجفي: هل يمكن للوكيل التنبؤ بالحالة التالية والمكافأة قبل بدء أي نشاط ما بعد التعلم؟ بمعنى آخر، تتمثل طريقة التمييز بين الخوارزميات القائمة على النموذج أو غير القائمة على النموذجفي فحص الخوارزميات ومعرفة ما إذاكانت تستخدم دوال التحويل والمكافأة. إذا تم استخدامها، فهي عبارة عن خوارزمية التعليم المعزز القائمة على النموذج.

كلتا الطريقتين لها نقاط قوة ونقاط ضعف. تضمن الطرق غير النموذجية إلى حد ما أنها ستعثر في النهاية على السياسة المثلى ولديها وقت حساب ضئيل جدًا لكل تجربة. ومع ذلك، فهم يستخدمون البيانات أثناء اختبار غير فعال للغاية، وبالتالي غالبًا ما يتطلبون قدرًا كبيرًا من الخبرة لتحقيق أداء جيد.في المقابل، يمكن للخوارزميات القائمة على النموذج التغلب على هذه المشكلة، لكن الوكيل يتعلم فقط من أجل نموذج معين وأحيانًا لا يكون مناسبًا لبعض النماذج الأخرى. يستغرق الأمر أيضًا وقتًا لتعلم نموذج آخر.

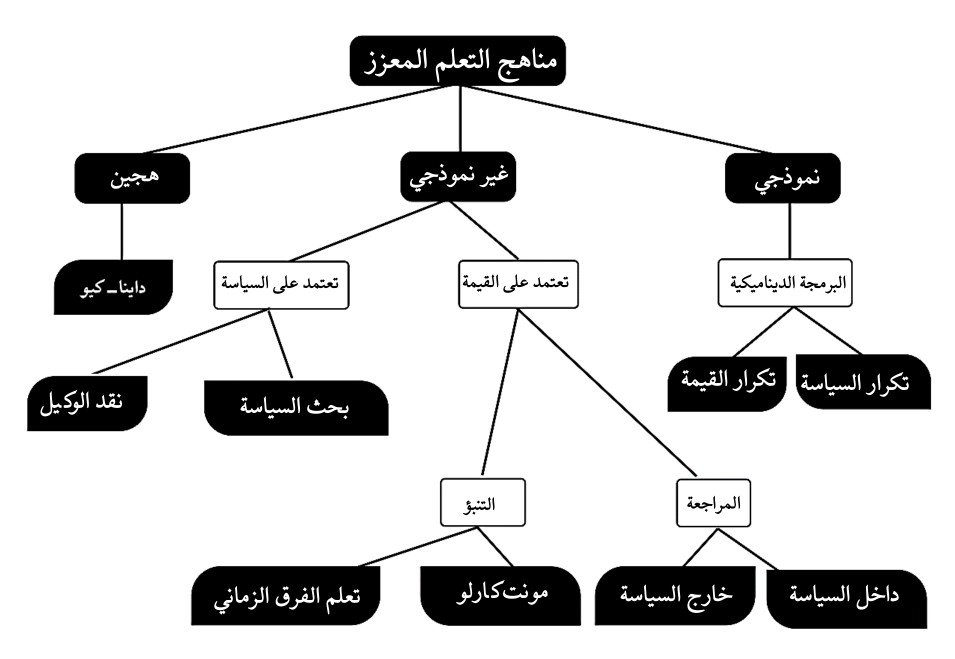
من المزايا المهمة لامتلاك نموذج أنه يسمح للوكيل بالتخطيط للمستقبل ومعرفة ما سيحدث لمجموعة واسعة من الإجراءات المختلفة التي يمكنه القيام بهافي حالته الحالية. ثم قارن هذه النتائج عندما تقرر التصرف. يمكن أن يؤدي هذا إلى تحسن كبيرفي الأداء مقارنة بالخوارزميات التي لا تستخدم النموذج. تعتمد الخوارزميات التي لا تعتمد على النموذج فقط على "التجربة

والخطأ" لتحديث معارفهم. لذلك، لا يحتاجون إلى تخزين جميع مجموعات الحالات والإجراءات.

يمكن تصنيف النهج غير النموذجي على أنه **داخل السياسة**[1]**وخارج السياسة**[2]. تستخدم الأساليب داخل السياسة السياسة الحالية لإنشاء الإجراءات واستخدامها لتحديث السياسة الحالية.في حين أنه،في الأساليب الخارجة عن السياسة، يتم استخدام سياسة استكشافية مختلفة لإنشاء إجراءات مقارنة بالسياسة الناشئة.

المناهج الكلاسيكية للتعليم المعزز

الآن بعد أن أصبحت على دراية بالمفاهيم الأساسية للتعليم المعزز، سنقوم في هذا القسم بوصف الأساليب الكلاسيكية للتعليم المعززفي حل المشكلات. يمكن تقسيم هذه الأساليب إلى نموذجية، وغير نموذجية، ومجموعة من النهجين كما هو موضحفي الشكل 9ـ6. فيما يلي، سوف ندرس هذه الأساليب.



الشكل 9ـ6. كيفية تقسيم مناهج التعليم المعزز.

[1] on-policy
[2] off-policy

البرمجة الديناميكية[1]

يشير مصطلح البرمجة الديناميكية إلى مجموعة من الخوارزميات التي يمكن استخدامها لحساب السياسة المثلى وفقًا لنموذج كامل للبيئة كعملية ماركوف لاتخاذ القرار. الفكرة الرئيسية للبرمجة الديناميكية والتعليم المعزز بشكل عام هي استخدام دوال القيمة لتنظيم هيكل البحث عن سياسات جيدة. خوارزميات البرمجة الديناميكية الكلاسيكية لها تطبيق محدودفي التعليم المعزز بسبب افتراض نموذج كامل وأيضًا بسبب التكلفة الحسابية العالية. ومع ذلك، لا تزال هذه الأساليب مهمة من الناحية النظرية.

تتضمن البرمجة الديناميكية نسختين مختلفتين من كيفية تنفيذها: **تكرار السياسة وتكرار القيمة**. فيما يلي، سوف نصف بإيجاز هذين النهجين.

تكرار السياسة

عندما يتم تحسين سياسة π باستخدام v_π للحصول على سياسة $\acute{\pi}$ أفضل، يمكن حساب $v_{\acute{\pi}}$ وإعادة تحسينها للحصول على سياسة "π أفضل. الحصول على قواعد موحدة ودوال القيمة:

$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} v_*,$$

إنه مضمون أن كل سياسة سيتم تحسينها عن السياسة السابقة؛ ما لم يكن هو بالفعل الأمثل. نظرًا لأن عملية صنع القرارفي ماركوف المحدود لا تحتوي إلا على عدد محدود من السياسات، يجب أن تتقارب هذه العملية مع سياسة واحدة ودالة القيمة المثلىفي عدد محدود من التكرارات.

تسمى هذه الطريقة للعثور على السياسة المثلى تكرار السياسة. وتجدر الإشارة إلى أن كل تقييم للسياسة، والذي يعد بحد ذاته حسابًا تكراريًا، يبدأ بدالة قيمة للسياسة السابقة. وعادة ما يؤدي هذا إلى زيادة هائلةفي سرعة تقارب تقييم السياسات؛ ربما لأن دالة القيمة تتغير قليلاً من سياسة إلى أخرى.

تكرار القيمة

تتمثل إحدى عيوب طريقة تكرار السياسةفي أن كل تكرار يتضمن تقييمًا للسياسة، والذي قد يتضمنفي حد ذاته حسابًا متكررًا طويلاً يتطلب تحولات متعددةفي مجموعة الحالات.في حالة تكرار تقييم السياسة، يحدث التقارب تمامًا مع v_π فقط ضمن النطاق المسموح به. نتيجة لذلك السؤال الذي يطرح نفسه، هل يجب أن ننتظر تقاربًا دقيقًا أم يمكننا التخلي عنه؟

يمكن تقصير خطوة تقييم السياسة، وتكرار السياسة بعدة طرق دون فقدان ضمان تقارب تكرار السياسة. تعتبر الحالة الخاصة مهمة عندما يتم إيقاف تقييم السياسة بعد خطوة واحدة فقط.

---

[1] Dynamic programming

تسمى هذه الخوارزمية تكرار القيمة، والتي يمكن كتابتها كعملية دعم بسيطة تجمع بين تحسين السياسة وخطوات تقييم السياسة:

$$v_{k+1}(s) = max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) | S_t = s, A_t = a]$$

$$= max_a \sum_{ś,r} p(ś, r | s, a)[r + \gamma v_k(ś)]$$

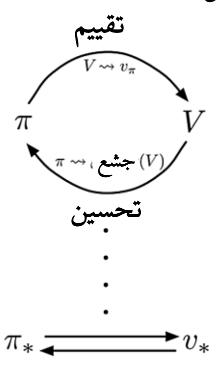
لأي v_0 مرغوب فيه ، يمكن إظهار أن التسلسل $\{v_k\}$ يمكن أن يتقارب مع v^* في ظل نفس الظروف التي تضمن وجود v^*.

أخيرًا ، لنتأمل كيف ينتهي تكرار القيمة. مثل تقييم السياسة ، يتطلب تكرار القيمة رسميًا عددًا غير محدود من التكرارات لتتقارب تمامًا من الناحية العملية ، ونتوقف عندما تتغير دالة القيمة بمقدار صغير فقط في نوبة واحدة.

تكرار خط السياسة المعمم[1]

يتضمن تكرار السياسة عمليتين متزامنتين ومتبادلتين، واحدة تجعل دالة القيمة متوافقة مع السياسة الحالية (تقييم السياسة)، والأخرى جشع للسياسة فيما يتعلق بدالة القيمة الحالية (تحسين السياسة).في تكرار السياسة، تكون هاتان العمليتان متناوبتين وتكتملان قبل أن تبدأ العملية الأخرى، لكن هذا ليس ضروريًا حقًا. على سبيل المثال،في عمليات تكرار القيمة، يتم إجراء تكرار واحد فقط لتقييم السياسة بين كل تحسين للسياسة.

يستخدم مصطلح تكرار السياسة المعمم للإشارة إلى الفكرة العامة للسماح بالتفاعل بين عمليات تقييم السياسة وتحسين السياسة، بغض النظر عن تفاصيل العمليتين. يظهر المخطط العام لتكرار السياسة المعممفي الشكل 9ـ7.



الشكل 9ـ7. تكرار السياسة المعمم. تتفاعل دوال القيمة والسياسة طالما أنها مثالية وبالتالي متوافقة.

[1] generalized policy iteration (GPI)

من السهل أن نرى أنهفي حالة استمرار كل من عملية التقييم وعملية التحسين، أي أنهما لم يعدا يتغيران، فيجب أن تكون دالة القيمة والسياسة هي الأمثل. تصبح دالة القيمة مستقرة فقط عندما تكون متوافقة مع السياسة الحالية، وتصبح السياسة مستقرة فقط عندما تكون جشعة فيما يتعلق بدالة القيمة الحالية. لذلك، يتم تثبيت كلتا العمليتين فقط عندما يتم العثور على سياسة جشعة لدالة التقييم الخاصة بها. يوضح هذا أن معادلة تحسين بلمان ثابتة، وبالتالي فإن السياسة والقيمة هما القيمتان الأمثل.

يمكن اعتبار عمليات التقييم والتحسينفي تكرار السياسة العامة بمثابة منافسة وتعاون. على المدى الطويل، تتفاعل هاتان العمليتان لإيجاد حل مشترك: دالة القيمة المثلى والسياسة المثلى.

مونت كارلو

على عكس البرمجة الديناميكية، تتعلم طريقة مونت كارلو من التجربة فقط. من نواح كثيرة، يمكن اعتباره أبسط طريقة للتعليم المعزز. تحدد طريقة مونت كارلو قيمة الحالة بتمريرها عدة مرات ومتوسط إجمالي المكافأة المستلمة بعد الموافقة على الحالة. نظرًا لأن إجمالي المكافأة المستلمة قبل إكمال المرحلة غير معروف، فإن طريقة مونت كارلو تنطبق فقط على مرحلة الاجراء. لا يتم التحديث إلا بعد انتهاء الحركةفي مساحة الحالة، والتي تسمى التحديث غير المتصل (التحديث أثناء التنقلفي مساحة الحالة يسمى التحديث الاونلاين). بالإضافة إلى ذلك،في طريقة مونت كارلو، تستند القيم إلى التجربة الفعلية، وليس على قيم الحالات البديلة.

تعتمد طرق مونت كارلو على فكرة تكرار السياسة المعمم. كما ذكرنا سابقًا، يعد تكرار السياسة المعمم مخططًا تكراريًا ويتكون من خطوتين.في الخطوة الأولى، نحاول إنشاء تقريب دالة القيمة بناءً على السياسة الحالية، والتي تُعرف باسم خطوة تقييم السياسة.في الخطوة الثانية، يتم تحسين السياسة وفقًا لدالة القيمة الحالية، والتي تُعرف بخطوة تحسين السياسة.في طريقة مونت كارلو، لتقدير دالة القيمة، يتم تنفيذ البرامج عن طريق تشغيل السياسة الحالية على النظام. يتم استخدام المكافأة التراكمية طوال المرحلة وتوزيع الحالات التي تمت مواجهتها لتشكيل تقدير لدالة القيمة. بعد ذلك، يتم تقدير السياسة الحالية بجشع وفقًا لدالة القيمة الحالية. باستخدام هاتين الخطوتين بشكل متكرر، يمكن إظهار أن الخوارزمية تتقارب مع القيمة المثلى ودالة السياسة. على الرغم من أن طرق مونت كارلو سهلة التنفيذ، إلا أنها تتطلب عددًا كبيرًا من التكرارات لتتقارب وتعاني من تباين كبيرفي تقدير دالة القيمة.

تعلم الفرق الزمني

يتضمن تعلم الفرق الزمني أفكار البرمجة الديناميكية ومونت كارلو. يقترب نهج الفرق الزمني من قيمة زوج الإجراء_الحالة من خلال مقارنة التقديرات عند نقطتينفي الوقت المناسب، ومن هنا جاء اسم الفرق الزمني. مثل البرمجة الديناميكية، تتعلم خوارزميات تعلم فرق الوقت تقدير القيم

بناءً على تقديرات أخرى (تسمى البوتسترابينك[1]). أيضًا، يمكن تعلم الفرق الزمني، مثل طريقة مونت كارلو، مباشرة من التجربة دون معرفة مسبقة بالبيئة. هذا يعني أن تعلم الفرق الزمني هو نهج تعليمي غير نموذجي (أو يمكن معادلته بالتعلم غير الخاضع للإشراف).

الفكرة الرئيسية لمنهج تعلم الفرق الزمني هي التعلم بناءً على الفرق بين تنبؤات الوقت المتتالية، ولا داعي للانتظار حتى نهاية المسار للتحديث. بمعنى آخر، الغرض من التعلم هو جعل التنبؤ الحالي للمتعلم لنمط الإدخال الحالي أكثر اتساقًا مع التنبؤ التاليفي الخطوة التالية. تكمن فكرة هذه الطريقةفي أنه بعد ملاحظة بعض المكافآت التي حصل عليها الوكيل بعد زيارة حالة ما وتنفيذ إجراء معين، فمن الممكن تقديم تقدير أفضل لمقدار زوج بين الحالة والاجراء.

أساليب مونت كارلو والبرمجة الديناميكية لها نقاط ضعف خطيرة؛ مونت كارلو غير قادر على حل المشكلات المستمرة، وتتطلب البرمجة الديناميكية نموذجًا للبيئة. يستخدم تعلم الفرق الزمني عينة احتياطيةفي طريقة مونت كارلو والتمهيدفي طريقة البرمجة الديناميكيةفي نفس الوقت. لذلك، يتم الجمع بين الأجزاء الجيدة فقط من الطرق السابقة.

تستخدم طريقة مونت كارلو المكافأة الإجمالية لتحديث دالة القيمة. حيث إن تعلم الفرق الزمني، بدلاً من تحديث القيمفي نهاية المرحلة (باستخدام المكافأة الإجمالية)، تقوم بتحديث القيم اونلاينفي كل مرحلة.

يمكن استخدام مخطط الدعم لمقارنة الأساليب الثلاثة ، مونت كارلو ، البرمجة الديناميكية ، وتعلم الفرق الزمني. يمكن رؤية مقارنة بين هذه الأساليب الثلاثة بناءً على مخطط الدعم للحالة والقيمة في الشكل 9ـ7.

غالبًا ما يشير تعلم الفرق الزمني إلى مشكلة تنبؤ بقاعدة تحديث لدالة قيمة معينة:

$$V(s) \leftarrow V(s) + \alpha\big(r + \gamma V(\acute{s}) - V(s)\big)$$

حيث α هو معدل التعلم و γ هو عامل الهبوط. يُعرف الجزء الموجود بين قوسين باسم خطأ الفرق الزمني:

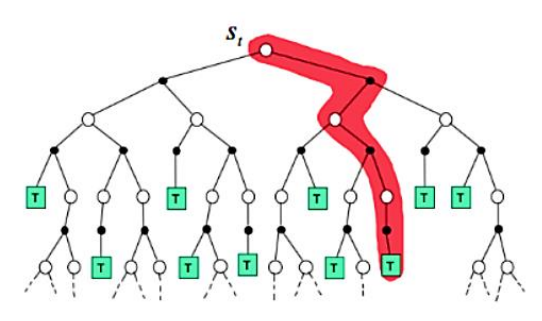
$$\delta_t = r + \gamma V(\acute{s}) - V(s)$$

تُستخدم طريقة اتعلم الفرق الزمني للتنبؤ بطريقتين مختلفتين للمراجعة. والفرق الرئيسي بين الطريقتين هو أن إحداهما في السياسة والأخرى خارج السياسة. الخوارزميتان المستخدمتان على نطاق واسع في التعليم المعزز هما Sarsa و Q-Learning. في ما يلي ، سوف نفحص هذين النوعين من خوارزميات الفرق الزمني.

[1] bootstrapping

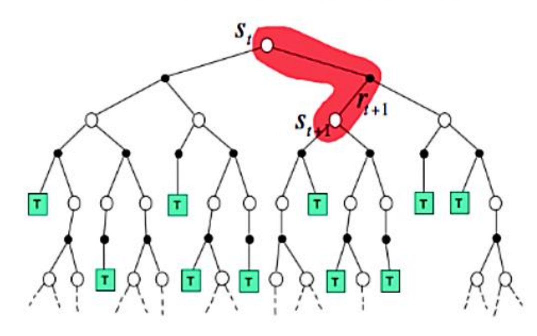
مونت كارلو

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$



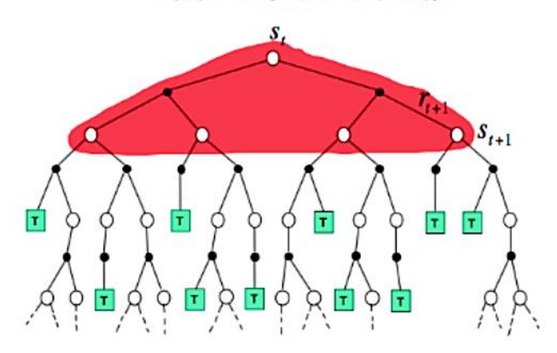
الفرق الزمني

$$V(S_t) \leftarrow V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$$



البرمجة الديناميكية

$$V(S_t) \leftarrow \mathbb{E}_\pi\left[R_{t+1} + \gamma V(S_{t+1})\right]$$



الشكل 9ـ7 مقارنة بين مخططات دعم مونت كارلو وتعلم الفرق الزمني والبرمجة الديناميكية لدوال الحالةـ القيمة

Sarsa (في السياسة)

Sarsa هي طريقة تعليم معزز بالفرق الزمني تتعلم الإجراء بدلاً من دالة القيمةفي محاولة لتقدير دالة القيمة. تمت تسمية هذا الميراث باسم: ("الحالة"، "الإجراء"، "المكافأة"، "الحالة التالية"، "الإجراء التالي"). هذا النهجفي السياسة، لأن يقدر $Q_\pi(s,a)$ لـ π للسياسة الحالية. قاعدة تحديث الحالةــ القيمة هي كما يلي:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma Q(\acute{s},\acute{a}) - Q(s,a)\big)$$

يمكن إثبات أن Sarsa يتقارب مع السياسة المثلى عند زيارة جميع أزواج غير محددة من الاجراءــ الحالة.

Q-Learning (خارج السياسة)

يُطلق على تعلم الفرق الزمني خارج سياسة بـ Q-Learning، وهو أحد أكثر الأساليب الأساسية والأكثر شيوعًا لتقدير دوال قيمة Q بطريقة غير نموذجية. قاعدة تحديث هذه الطريقة هي كما يلي:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma max_{\acute{a}} Q(\acute{s},\acute{a}) - Q(s,a)\big)$$

تحاول Q-Learning بذل قصارى جهدها في الوضع الحالي. تعتبر هذه الخوارزمية خارج السياسة. وذلك لأن دالة Q-Learning تتعلم من الإجراءات خارج السياسة الحالية. بشكل عام، تسعى Q-Learning إلى تعلم السياسة من أجل تعظيم المكافأة الإجمالية. اليوم ، تعتمد العديد من خوارزميات التعلم العميق المعزز على Q-Learning.

Q-Learning غير مبال بالاستكشاف. وهذا يعني أنه بغض النظر عن السياسة الاستكشافية المتبعة ، فإنها تتقارب مع السياسة المرغوبة ؛ بافتراض أن كل زوج من إجراءات الحالة تمت زيارته عددًا غير محدود ويتم تقليل معامل التعلم α بشكل مناسب.

البحث فى السياسة

لا تحتاج طرق البحثفي السياسة إلى الحفاظ على نموذج دالة القيمة، ولكنها تبحث مباشرة عن السياسة المثلى π^*. من بين خوارزميات البحث عن السياسةفي التعلم المعزز، تعد خوارزمية التدرج السياسي هي الأكثر شيوعًا.

الغرض من التعلم المعزز هو إيجاد استراتيجية سلوك مثالية للوكيل للحصول على المكافآت المثلى. تعمل طرق التدرج السياسي مباشرة على نمذجة السياسة وتحسينها. يتم تقييم هذه السياسة من خلال تنفيذ الإجراءات المنصوص عليهافي السياسة الحالية وحساب المكافأة. ثم يتم تحديث معاملات السياسة لزيادة العائد المتوقع باستخدام التدرج التنازلي. يمكن كتابة قاعدة التحديث لمعاملات السياسة على النحو التالي وفقًا للعائد المتوقع J:

$$\theta_{i+1}^{\pi} = \theta_{i}^{\pi} + \alpha \nabla_{\theta^{\pi}} j \ \ , j = \mathbb{E}_{\pi}(\sum_{k=0}^{\infty} \gamma^{k} r_{k})$$

البحثفي السياسة لديه تقارب أفضل ويمكن أن يتعلم سياسات عشوائية غير ممكنة مع الأساليب القائمة على القيمة. العيب الرئيسي لخوارزميات السياسة هو مرحلة تقييم السياسة، والتي تعاني من تباين كبير وبالتالي يمكن أن تكون بطيئةفي تعلم السياسات الجيدة.

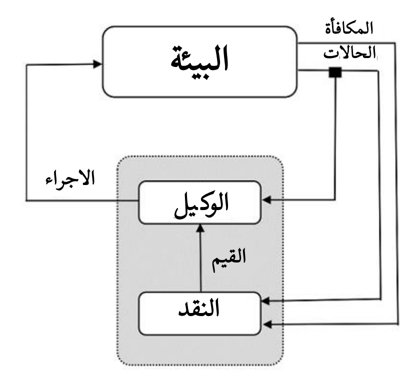
نقد الوكيل

طرق التدرج السياسي النقي بطيئة التعلم بسبب تقديرات التباين العالية وغير مناسبة لأداء المشكلات الاونلاين. ومع ذلك، يمكن استخدام طرق الفرق الزمني للتعامل مع هذه المشاكل.

تتطلب مقاربات دالة القيمة نظريًا التغطية الكاملة لمساحة الحالة والقيم المضخمة المقابلة لجميع العمليات الممكنةفي كل حالة. لذلك، عند العمل مع التطبيقات عالية الأبعاد، فإنه يحتوي على الكثير من التعقيد الحسابي، وأيضًا، قد يتسبب تغيير بسيطفي القيم المضخمة المحليةفي حدوث تغيير كبيرفي السياسة.

على عكس طرق دالة القيمة ، تراعي طرق بحث السياسة السياسة الحالية والسياسة التالية للسياسة الحالية ، ثم تحسب التغييرات في معاملات السياسة ، مما يؤدي إلى تعقيد حسابي أقل بكثير من أساليب دالة القيمة. ومع ذلك ، قد تؤدي مناهج بحث السياسة إلى التحسين المحلي وتفشل في تحقيق التحسين العالمي.

من خلال الجمع بين هذين النهجين، تتعلم طرق نقد الوكيلفي نفس الوقت سياسة ودالة القيمة_الحالة. يُعرف هذا النهج، الذي يتم تقديمه من خلال الجمع بين دالة القيمة ونهج البحث عن السياسة للاستفادة من كلتا الطريقتين، باسم هيكل نقد الوكيل. يمكن القول إن طريقة نقد الوكيل هي طريقة تعلم الفرق الزمني التي تخزن السياسة بشكل صريح. كما هو مبينفي الشكل 9_8، يتم اختيار الإجراء بواسطة وكيل التحكم ويتم استخدام النقد لنقل القيم إلى الوكيل. لذلك، يعتمد قرار تحديث السياسة على هذا النقد.



الشكل 9_8. هيكل نقد الوكيل

الطريقة المشتركة (داينا-كيو[1])

داينا_كيو هي إحدى البنى التي لديها القدرة على الجمع بين القدرة على التعلم بطريقة غير نموذجية والقدرة على البرمجةفي الأساليب القائمة على النموذج. تتعلم الخوارزمية نموذجًا من خلال رؤية الحالات التي تتم زيارتها والمكافآت التي تحصل عليهافي حالة معينة، وتستخدم المعلومات لتحديث احتمالات التحويل ودالة المكافأة.

التعلم بالمحاكاة[2]

على الرغم من أن التعليم المعزز قد حقق بعضًا من أكثر التطورات التقليدية في الذكاء الاصطناعي في السنوات الأخيرة ، إلا أن نجاحه اقتصر إلى حد كبير على مجالات المحاكاة حيث يتم تحديد إشارة المكافأة وإعطاءها جيدًا. على الرغم من أنه قد يكون واضحًا بشكل بديهي ما هي دالة المكافأة في المجالات المحاكاة (مثل النقاط في ألعاب Atari) ، في العديد من مجالات العالم الحقيقي ، من الصعب جدًا افتراض إمكانية الوصول إلى دالة المكافأة. ضع في اعتبارك مهمة تعلم سياسة قيادة جيدة. بينما يمكن للسائق البشري الجيد القيادة بأمان على الطريق ، فقد لا يكون قادرًا على صياغة دالة المكافأة رياضياً والتي تميز بدقة بين سياسات القيادة الجيدة والسيئة. بدون أداء جيد للمكافأة ، فإن التعليم المعزز غير مناسب لحل مشكلة القيادة الذاتية.

على الرغم من صعوبة تحديد دالة المكافأة ، يمكن تعلم سياسة جيدة من خلال التقليد المباشر للمسارات التي يوفرها خبير قد يكون أو لا يكون لديه إمكانية الوصول إلى دالة المكافأة الفعلية ؛ ومع ذلك ، فهو يعرف كيف يتصرف في بيئة مثالية. من الناحية المثالية ، لا يتذكر الخط مسارات الخبراء فحسب ، بل يعمم أيضًا سلوك الخبراء في الحالات غير المرئية على مسارات الخبراء. يُعرف هذا النهج لتعلم السياسة من خلال التقليد باسم **التعلم بالمحاكاة**.

الهدف من تقنيات التعلم بالمحاكاة هو محاكاة السلوك البشري في مهمة محددة. يتم تدريب الوكيل على أداء مهمة من خلال تعلم التعيين بين الملاحظات والإجراءات. كانت فكرة التعلم بالمحاكاة موجودة منذ سنوات. ومع ذلك ، فقد حظي هذا المجال بالاهتمام مؤخرًا بسبب التقدم في الحوسبة بالإضافة إلى زيادة الطلب على التطبيقات الذكية. إن نموذج التعلم عن طريق المحاكاة يكتسب شعبية ، لأنه يسهل تدريب المهام المعقدة بأقل قدر من المعرفة المتخصصة بالمهام. يمكن أن تقلل أساليب التعلم المقلدة من مشكلة تدريب مهمة ما لمشكلة عرضها دون الحاجة إلى برمجة صريحة أو تصميم دوال مكافأة محددة لتلك المهمة.

---

[1] Dina Q

[2] Imitation Learning

الفرق بين التعلم بالمحاكاة والتعلم المعزز

في التعلم بالمحاكاة ، يلاحظ المتعلم أولاً تصرفات الخبير (غالبًا ما يكون بشريًا) في مرحلة التعلم. ثم يستخدم المتعلم مجموعة التدريب هذه لتعلم السياسة ، والتي تحاول محاكاة الإجراءات التي أظهرها الخبير لتحقيق أفضل أداء. من ناحية أخرى ، لا توجد مثل هذه الخبرة في التعلم المعزز. بدلاً من ذلك ، يمتلك الوكيل دالة "مكافأة" ويستخدم استراتيجيات لاستكشاف الموقف ومساحة العمل بفعالية ، ويوفر وحده (عن طريق التجربة والخطأ) السياسة المثلى. لنفكر في الوكيل الذاتي. يمكن أن تكون مساحة الحالة هي ما يراه الوكيل في أي لحظة: الطريق ، والإشارات ، والمركبات الأخرى ، والمشاة. يمكن أن تكون الإجراءات هي التوجيه والتسارع والتوقف.

عامل التعلم بالمحاكاة يراقب السائق البشري (الخبير) ويسجل أفعاله في المواقف المختلفة. ومن ثم ، فإنه ينشئ سياسة في كل موقف ، بناءً على ما قام به الخبير ، والإجراءات التي ينبغي اتخاذها. في وقت التشغيل ، يبذل قصارى جهده لتقريب الإجراء الصحيح بناءً على السياسة ، لأن المواقف ليست متطابقة تمامًا وسيتم تقديم عنصر الاحتمال.

في المقابل ، يتلقى عامل التعلم المعزز دالة المكافأة. على سبيل المثال ، يتم احتساب كل ثانية تمر كنقطة إيجابية ، وإذا اصطدمت أو اصطدمت بمشاة أو سيارة أخرى ، تنتهي المهمة بمكافأة صفرية. يبدأ الوكيل بعد ذلك بإجراءات عشوائية ويتعلم من خلال التجربة والخطأ الإجراءات التي تزيد من المكافأة وأيها تصل إلى الصفر. بعد فترة ، عندما يكون السياسة جيدًة بما تكفي (بناءً على معايير الأداء) ، يستخدم الوكيل السياسة للقيادة. لذا فإن كلا الطريقتين تؤديان إلى إستراتيجية وقت التشغيل. فقط طريقة تعلم السياسة مختلفة.

التعلم عن طريق المحاكاة المباشرة (محاكاة السلوك)

تواجه عملية اتخاذ القرار متعددة الخطوات في التعلم المعزز مساحة بحث ضخمة ، لذلك ليس من السهل استخدام المكافآت التراكمية لتعلم القرارات الصحيحة التي تم اتخاذها عدة خطوات مسبقًا. ومع ذلك ، يمكن الحد من مثل هذه المشكلة بشكل جيد عن طريق المحاكاة المباشرة لأزواج الحالة_الاجراء التي يقدمها الخبراء البشريون ، وهو ما يسمى **التعلم المحاكي المباشر (المعروف باسم محاكاة السلوك)**.

لنفترض أن الخبراء البشريين قد أعطانا مجموعة من المسارات $\{\tau_1, \dots, \tau_m\}$ ، حيث يحتوي كل مسار على سلسلة من الحالات والإجراءات.:

$$\tau_i =< s_1^i, a_1^i, s_2^i, a_2^i, \dots, s_{n_i+1}^i >$$

حيث n_i هو عدد الانتقالات في المسار i. هذه المعلومات تخبر الوكيل بما يجب فعله في كل حالة. وبالتالي ، يمكننا استخدام التعلم الخاضع للإشراف لتعلم السياسة التي تتوافق مع بيانات المسار التي تم الحصول عليها من الخبراء البشريين.

يمكننا استخراج أزواج الحالة_الإجراء من جميع المسارات لإنشاء مجموعة بيانات جديدة:

$$D = \{(s_1, a_1), (s_2, a_2), \dots, (s_{\sum_{i=1}^{m} n_i}, a_{\sum_{i=1}^{m} n_i})\}$$

الذي يعلق على الحالات كسمات والإجراءات كعلامات.

بعد ذلك ، من مجموعة البيانات D ، يمكننا تعلم نموذج سياسة باستخدام خوارزميات التصنيف للاجراءات المنفصلة أو خوارزميات الانحدار للاجراءات المستمرة.

ثم تقوم محاكاة السلوك (BC) بحل السياسة عن طريق تقليل الخسارة التعليمية الخاضعة للإشراف:

$$min_{\pi} J_{BC}(\pi) = -\frac{1}{N} \sum_{k=1}^{N} \log \pi(s_k, a_k)$$

التعلم المعزز العكسي[1]

في العديد من التطبيقات ، غالبًا ما يكون تصميم دالة المكافأة أمرًا صعبًا ، ولكن قد نتمكن من اشتقاق دالة المكافأة بشكل عكسي من الأمثلة التي قدمها الخبراء البشريون. هذا النهج يسمى **التعلم المعزز العكسي**.

في التعلم المعزز العكسي ، تُعرف مساحة الحالة X ومساحة الإجراء A ، بالإضافة إلى مجموعة بيانات المسار $\{\tau_1, \dots, \tau_m\}$، تمامًا كما لدينا في التعلم بالمحاكاة المباشر. الفكرة الرئيسية للتعلم المعزز العكسي هي كما يلي:

إن السماح للوكيل بتنفيذ الإجراءات التي تتوافق مع الأمثلة المقدمة يكافئ إيجاد السياسة المثلى في بيئة دالة المكافأة ، حيث تخلق السياسة المثلى نفس المسارات مثل الأمثلة المقدمة. بمعنى آخر ، نحن نبحث عن دالة المكافأة التي تكون الأمثلة المقدمة هي الأمثل ، ثم نستخدم دالة المكافأة هذه لتعليم سياسة التعلم المعزز.

افترض أنه يمكن تمثيل دالة المكافأة كدالة خطية للحالات ، أي $R(x) = w^T x$. بعد ذلك ، يمكن كتابة المكافآت التراكمية لسياسة π على النحو التالي:

$$\rho^{\pi} = \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t R(x_t) \,|\pi\right] = \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t w^T x_t \,|\pi\right]$$
$$= w^T \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t x_t \,|\pi\right].$$

وهو ناتج الضرب الداخلي للمعاملات w والقيمة المتوقعة لمجموع اوزان متجهات الحالة.

---

[1] Inverse Reinforcement Learning

افرض $\bar{x}^{\pi}$ تمثل القيمة المتوقعة لمتجهات الحالة $\mathbb{E}[\sum_{t=0}^{+\infty}\gamma^{t}x_{t}\ |\pi]$. يمكن تقريب القيمة المتوقعة $\bar{x}^{\pi}$ باستخدام طريقة مونت كارلو:

يمكن رؤية أمثلة المسارات كعينات من السياسة المثلى، لذلك يمكننا حساب الوزن الإجمالي للحالات في كل مثال مسار ثم أخذ المتوسط، المشار إليه بـ $\bar{x}^{*}$. بعد ذلك، بالنسبة لدالة المكافأة المثلى $R(x) = {w^{*}}^{T}x$، لدينا القيمة المتوقعة $\bar{x}^{\pi}$ الناتج عن أي سياسة أخرى:

$${w^{*}}^{T}\bar{x}^{*} - {w^{*}}^{T}\bar{x}^{\pi} = {w^{*}}^{T}(\bar{x}^{*} - \bar{x}^{\pi}) \geq 0.$$

إذا استطعنا حساب $(\bar{x}^{*} - \bar{x}^{\pi})$ لكل سياسة، فيمكننا ذلك حل المسئلة.

$$w^{*} = argmax_{w}\ min_{\pi} w^{T}(\bar{x}^{*} - \bar{x}^{\pi})$$
$$s.t. \|w\| \leq 1.$$

نظرًا لصعوبة الحصول على جميع السياسات، فإن النهج البديل هو البدء بسياسة عشوائية واستخدامها للعثور على دالة أفضل للمكافأة مرارًا وتكرارًا. ثم يتم استخدام دالة المكافأة لإيجاد سياسة أفضل. تستمر هذه العملية طالما لدينا دالة المكافأة والسياسة المناسبة لمسارات العينة.

باختصار، في التعلم المعزز العكسي، نبدأ بمجموعة من عروض الخبراء (بافتراض أنها مثالية)، ثم نحاول تقدير دالة المكافأة البارامترية، التي تسبب سلوك الخبير (السياسة).

خلاصة الفصل التاسع

- المبدأ الأساسي وراء النماذج الجماعية هو أن مجموعة من المتعلمين الضعفاء يجتمعون لمساعدة بعضهم البعض في تكوين متعلم قوي.
- لن يعمل أي نموذج للتعلم الآلي وحده بشكل أفضل على جميع أنواع البيانات ومجموعات البيانات.
- في نظرية التعلم الجماعي، يُطلق على المتعلمين الضعفاء (أو النماذج الأساسية) نماذج يمكن استخدامها ككتل لتصميم نماذج أكثر تعقيدًا من خلال الجمع بين عدة نماذج.
- التجميع هو طريقة بسيطة لكنها فعالة لإنشاء مجموعة من الفئات.
- الفكرة من وراء التجميع هي الجمع بين نتائج عدة نماذج للحصول على نتيجة إجمالية.
- التجميع يعمل على أساس أخذ عينات Bootstrap.
- في التمكين، تتأثر كل فئة بأداء الفئات التي تم إنشاؤها قبل إنشائها.
- التعلم مدى الحياة هو نموذج متقدم للتعلم الآلي يتعلم باستمرار، ويجمع المعرفة المكتسبة في العمل السابق، ويستخدمها للمساعدة في التعلم في المستقبل.
- غالبًا ما يتم تدريب الشبكات العصبية العميقة على افتراضات العالم المغلق، أي أنه يُفترض أن يكون توزيع البيانات التجريبية مشابهًا لتوزيع بيانات التدريب.
- عندما تعالج الشبكات العصبية العميقة البيانات التي لا تشبه التوزيع الذي لوحظ أثناء التدريب، فإنها غالبًا ما تقدم تنبؤات خاطئة وتقوم بذلك بثقة كبيرة.

- ينبغي ، إن أمكن ، تعميم نظام الذكاء الاصطناعي المثالي على عينات خارج التوزيع.
- يعد الكشف عن التوزيع غير ضروري لضمان موثوقية وسلامة أنظمة التعلم الآلي.
- يحاول التعلم المعزز من خلال الوكيل حل المشكلة عن طريق التجربة والخطأ من خلال التفاعل مع بيئة غير معروفة للوكيل.
- عمليات صنع القرار في ماركوف هي نموذج رياضي عشوائي لسيناريو صنع القرار.

مصادر إضافية لمزيد من القراءة

Vitay, J. (2020). Deep Reinforcement Learning.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Sugiyama, M. (2015). Statistical reinforcement learning: modern machine learning approaches. CRC Press.

Zhou, Z. H. (2019). Ensemble methods: foundations and algorithms. Chapman and Hall/CRC.

المصادر

میلاد وزان، یادگیری عمیق: اصول، مفاهیم و رویکردها، ویرایش نخست، تهران، میعاد اندیشه، ۱۳۹۹

Abe, S. (2005). Support vector machines for pattern classification (Vol. 2, p. 44). London: Springer.

Aljundi, R. (2019). Continual learning in neural networks. arXiv preprint arXiv:1910.02718.

Alpaydin, E. (2004). Introduction To Machine Learning. S.L.: Mit Press.

Andrew Carlson, Justin Betteridge, and Bryan Kisiel. (2010). Toward an Architecture for Never-Ending Language Learning. In AAAI, pages 1306-1313.

Anthony Aue and Michael Gamon. (2005). Customizing Sentiment Classifiers to New Domains: A Case Study. RANLP.

Avrim Blum. (1998). On-line algorithms in machine learning. Springer.

BENDALE, A., AND BOULT, T. (2015). Towards open world recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Bishop, Christopher M (2006). Pattern recognition and machine learning. springer.

Bonner, R., On Some Clustering Techniques. IBM journal of research and development.

Brudfors, M. (2020). Generative Models for Preprocessing of Hospital Brain Scans (Doctoral dissertation, UCL (University College London)).

Charte, F., Rivera, A. J., & Del Jesus, M. J. (2016). Multilabel classification: problem analysis, metrics and techniques. Springer International Publishing.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected CRFs. In Proceedings of the Third International Conference of Learning Representations (ICLR).

Chen, Z. (2016). Lifelong Machine Learning for Topic Modeling and Classification (Doctoral dissertation, University of Illinois at Chicago).

Chen, Z., & Liu, B. (2018). Lifelong machine learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 12(3), 1-207.

Chinnamgari, S. K. (2019). R Machine Learning Projects: Implement supervised, unsupervised, and reinforcement learning techniques using R 3.5. Packt Publishing Ltd.

Cremer, C. (2020). Approximate Inference in Variational Autoencoders (Doctoral dissertation, University of Toronto (Canada)).

David Lopez-Paz and Marc'Aurelio Ranzato. (2017). Gradient episodic memory for continual learning. In Advances in Neural Information Processing Systems (NeurIPS.

Ditterrich, T. (1997). Machine learning research: four current direction, Artificial Intelligence Magzine 4: 97–136
Du, K. L., & Swamy, M. N. (2013). Neural networks and statistical learning. Springer Science & Business Media.
Dulhare, U. N., Ahmad, K., & Ahmad, K. A. B. (Eds.). (2020). Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications. John Wiley & Sons.
Dutt, Anuvabh, Denis Pellerin, and Georges Quenot (2019). "Coupled ensembles of neural networks". Neurocomputing.
Edwin V Bonilla, Kian M Chai, and Christopher Williams. (2008). Multitask Gaussian Process Prediction. NIPS.
Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). Learning from imbalanced data sets (Vol. 10, pp. 978-3). Berlin: Springer.
Friedemann Zenke, Ben Poole, and Surya Ganguli. (2017). Continual learning through synaptic intelligence. In International Conference on Machine Learning (ICML).
Gan, Z. (2018). Deep Generative Models for Vision and Language Intelligence (Doctoral dissertation, Duke University).
Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), volume 9.
Golvano, D. C., Wiering, M., & Kasaei, H. (2020). Hierarchical Reinforcement Learning Approach to Lifelong Learning.
Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press.
Goodfellow, Ian J et al. (2013). "Challenges in representation learning: A report on three machine learning contests". In: International Conference on Neural Information Processing. Springer
Gordon, J. Bayesian Deep Generative Models for Semi-Supervised and Active Learning.
Gupta, P., & Sehgal, N. K. (2021). Introduction to Machine Learning in the Cloud with Python: Concepts and Practices. Springer Nature.
Haiqin Yang, Zenglin Xu, Irwin King, and Michael R Lyu. (2010). Online learning for group lasso. In ICML, pages 1191-1198.
Hastie, T., Tibshirani, R., and Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.
He, Kaiming et al. (2016). "Deep residual learning for image recognition". Proceedings of the IEEE conference on computer vision and pattern recognition.
Hertzmann, A. and Fleet, D. (2012). Machine Learning and Data Mining Lecture Notes. Computer Science Department University of Toronto.
Isele, D. (2018). Lifelong Reinforcement Learning on Mobile Robots. University of Pennsylvania.
James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaum Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. (2017). Overcoming catastrophic

forgetting in neural networks. Proceedings of the National Academy of Sciences of the United States of America.
Jebara, T. (2012). Machine learning: discriminative and generative (Vol. 755). Springer Science & Business Media.
Jing Jiang. (2008). A literature survey on domain adaptation of statistical classifiers. Technical report.
Johnston, B., Jones, A., & Kruger, C. (2019). Applied Unsupervised Learning with Python: Discover hidden patterns and relationships in unstructured data with Python. Packt Publishing Ltd.
Jung, A. (2021). Machine Learning: The Basics. under preparation, 2021, available online at https://alexjungaalto. github. io/MLBasicsBook. pdf.
Justus, Daniel et al. (2018). “Predicting the Computational Cost of Deep Learning Models”. 2018 IEEE International Conference on Big Data (Big Data). IEEE
Kazak, V. (2019). Unsupervised feature extraction with autoencoder: for the representation of parkinson´ s disease patients (Doctoral dissertation).
Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). Fundamentals of machine learning for predictive data analytics: algorithms. Worked Examples, and Case Studies.
Kingma, D. P. (2017). Variational inference & deep learning: A new synthesis.
Lesort, T. (2020). Continual Learning: Tackling Catastrophic Forgetting in Deep Neural Networks with Replay Processes. arXiv preprint arXiv:2007.00487.
Li, F.-F., Johnson, J., and Yeung, S. (2017). CS231n: Convolutional neural networks for visual recognition
Maaløe, L., Winther, O., Nielsen, O. N., Hauberg, S., Paquet, U., & Turner, R. (2018). Deep Generative Models for Semi-Supervised Machine Learning.
Macko, Vladimir et al. (2019). “Improving neural architecture search image classifiers via ensemble learning”. In: arXiv preprint arXiv:1903.06236.
Maimon, O., & Rokach, L. (Eds.). (2005). Data mining and knowledge discovery handbook.
Marsland, S. (2015). Machine Learning : an algorithmic perspective. Boca Raton, Fl: Crc Press.
McNicholas, P. D. (2016). Mixture model-based classification. Chapman and Hall/CRC.
Mitchell, T.M. (1997). Machine learning. New York: Mcgraw Hill.
Moolayil, J. (2016). Smarter Decisions–The Intersection of Internet of Things and Decision Science. Packt Publishing Ltd.
Neculae, A. (2020). Ensemble learning for spiking neural networks (Doctoral dissertation, The University of Manchester (United Kingdom)).
Neto, U. M. B., & Dougherty, E. R. (2015). Error estimation for pattern recognition. John Wiley & Sons.
Patel, A. A. (2019). Hands-on unsupervised learning using Python: how to build applied machine learning solutions from unlabeled data. O'Reilly Media.
Peng, Hao et al. (2015). “A comparative study on regularization strategies for embeddingbased neural networks”. In: arXiv preprint arXiv:1508.03721.
Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with Deep Convolutional Generative Adversarial Networks. In

Proceedings of the Fourth International Conference of Learning Representations (ICLR).
Radford, Alec, Luke Metz, and Soumith Chintala (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks". arXiv preprint arXiv:1511.06434.
Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. (2017). Expert gate: Lifelong learning with a network of experts. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. 2007. Self-taught Learning : Transfer Learning from Unlabeled Data. ICML, pages 759-766.
Rhys, H. (2020). Machine Learning with R, the tidyverse, and mlr. Simon and Schuster.
Rich Caruana. (1997). Multitask Learning. Machine learning, 28(1):41-75.
Rokach, L. (2010). Pattern classification using ensemble methods (Vol. 75). World Scientific.
Rokach, L., & Maimon, O. Z. (2007). Data mining with decision trees: theory and applications (Vol. 69). World scientific.
Salimans, Tim and Durk P Kingma (2016). "Weight normalization: A simple reparameterization to accelerate training of deep neural networks". Advances in Neural Information Processing Systems.
Sebastian Thrun. (1998). Lifelong Learning Algorithms. In S Thrun and L Pratt, editors, Learning to Learn, pages 181{209. Kluwer Academic Publishers.
SHALEV-SHWARTZ, S., ET AL. (2012). Online learning and online convex optimization. Foundations and Trends in Machine Learning.
Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS)
Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". arXiv preprint arXiv:1409.1556
Sinno Jialin Pan and Qiang Yang. (2010). A Survey on Transfer Learning. IEEE Trans. Knowl. Data Eng., 22(10):1345{1359
Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. (2014). Striving for simplicity: The all convolutional net. CoRR, abs/1412.6806.
Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". The journal of machine learning research 15.1.
Steffen Bickel, Michael Br¨uckner, and Tobias Scheffer. (2007). Discriminative Learning for Differing Training and Test Distributions. ICML.
Sun, L., Ji, S., & Ye, J. (2019). Multi-label dimensionality reduction. Chapman and Hall/CRC.
Swamy, G., Choudhury, S., Bagnell, J. A., & Wu, S. (2021). Of moments and matching: A game-theoretic framework for closing the imitation gap. In International Conference on Machine Learning (pp. 10022-10032). PMLR.
Szegedy, Christian et al. (2016). "Rethinking the Inception Architecture for Computer Vision". The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

T Mitchell, W Cohen, E Hruschka, P Talukdar, J Betteridge, A Carlson, B Dalvi, M Gardner, B Kisiel, J Krishnamurthy, N Lao, K Mazaitis, T Mohamed, N Nakashole, E Platanios, A Ritter, M Samadi, B Settles, R Wang, D Wi jaya, A Gupta, X Chen, A Saparov, M Greaves, and J Welling. (2015). Never-Ending Learning. AAAI.

Tang, Y. (2015). Learning Generative Models Using Structured Latent Variables. University of Toronto (Canada).

Touvron, Hugo et al. (2019). “Fixing the train-test resolution discrepancy”. arXiv preprint arXiv:1906.06423.

Touvron, Hugo et al. (2019). “Fixing the train-test resolution discrepancy”. In: arXiv preprint arXiv:1906.06423.

Vazan, Milad & Razmara, Jafar. (2021). Jointly Modeling Aspect and Polarity for Aspect-based Sentiment Analysis in Persian Reviews. 10.13140/RG.2.2.12339.14887/1.

Vazan, Milad. (2021). A Novel Approach for Enhancing Sentiment Classification of Persian Reviews Using Convolutional Neural Network and Majority Voting Classifier.

Vazan, Milad. (2021). Joint Learning for Aspect and Polarity Classification in Persian Reviews Using Multi-Task Deep Learning.

Verbeek, J. (2004). Mixture models for clustering and dimension reduction (Doctoral dissertation, Universiteit van Amsterdam).

Williams, P. J. (2021). Ensemble learning through cooperative evolutionary computation (Doctoral dissertation, University of Otago).

Witten, I. H., Frank, E., Hall, M. A., Pal, C. J., & DATA, M. (2005). Practical machine learning tools and techniques. In DATA MINING (Vol. 2, p. 4).

XU, H., LIU, B., SHU, L., AND YU, P. S. (2018). Learning to accept new classes without training.

Yang, X. S. (2019). Introduction to algorithms for data mining and machine learning. Academic press.

Yu, F. and Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. In Proceedings of the Fourth International Conference of Learning Representations (ICLR).

Zhou Z. (2021). Machine Learning. Springer Nature Singapore Pte Ltd.

Machine Learning
and Data Science

Foundations, concepts, algorithms, and tools

Milad Vazan

Translated Into Arabic by

Dr. Alaa Taima

